SPARSE LARGE-SCALE MULTI-OBJECTIVE OPTIMIZATION FOR CLIMATE-SMART AGRICULTURAL INNOVATION By

Ian Meyer Kropp

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Biosystems Engineering — Doctor of Philosophy Computer Science — Dual Major

ABSTRACT

SPARSE LARGE-SCALE MULTI-OBJECTIVE OPTIMIZATION FOR CLIMATE-SMART AGRICULTURAL INNOVATION

By

Ian Meyer Kropp

The challenge of our generation is to produce enough food to feed the present and future global population. This is no simple task, as the world population is expanding and becoming more affluent, and conventional agriculture often degrades the environment. Without a healthy and functional environment, agriculture as we know it will fail. Therefore, we must equally balance our broad goals of sustainability and food production as a single system. Multi-objective optimization, algorithms that search for solutions to complex problems that contain conflicting objectives, is an effective tool for balancing these two goals. In this dissertation, we apply multi-objective optimization to find optimal management practices for irrigating and fertilizing corn.

There are two areas for improvement in multi-objective optimization of corn management: existing methods run burdensomely slow and do not account for the uncertainty of weather. Improving run-time and optimizing in the face of weather uncertainty are the two goals of this dissertation. We address these goals with four novel methodologies that advance the fields of biosystems & agricultural engineering, as well as computer science engineering.

In the first study, we address the first goal by drastically improving the performance of evolutionary multi-objective algorithms for sparse large-scale optimization problems. Sparse optimization, such as irrigation and nutrient management, are problems whose optimal solutions are mostly zero. Our novel algorithm, called sparse population sampling (SPS), integrates with and improves all population-based algorithms over almost all test scenarios. SPS, when used with NSGA-II, was able to outperform the existing state-of-the-art algorithms with the most complex of sparse large-scale optimization problems (i.e., 2,500 or more decision variables).

The second study addressed the second goal by optimizing common management practices in a study site in Cass County, Michigan, for all climate scenarios. This methodology, which relied on SPS from the first goal, implements the concept of innovization in agriculture. In our innovization framework, 30 years of management practices were optimized against observed weather data, which in turn was compared to common practices in Cass County, Michigan. The differences between the optimal solutions and common practices were transformed into simple recommendations for farmers to apply during future growing seasons. Our recommendations drastically increased yields under 420 validation scenarios with no impact on nitrogen leaching.

The third study further improves the performance of sparse large-scale optimization. Where SPS was a single component of a population-based algorithm, our proposed method, S-NSGA-II, is a novel and complete evolutionary algorithm for sparse large-scale optimization problems. Our algorithm outperforms or performs as well as other contemporary sparse large-scale optimization algorithms, especially in problems with more than 800 decision variables. This enhanced convergence will further improve multi-objective optimization in agriculture.

Our final study, which addresses the second goal, takes a different approach to optimizing agricultural systems in the face of climate uncertainty. In this study, we use stochastic weather to quantify risk in optimization. In this way, farmers can choose between optimal management decisions with full understanding of the risks involved in every management decision.

Copyright by IAN MEYER KROPP 2022 To Jonas and his generation. This is for you.

ACKNOWLEDGMENTS

This work would not be possible without the contributions of many talented researchers, family, and friends. Firstly, Dr. Nejadhashemi provided the original spark for this research back in 2017, which I had the privilege to develop into this dissertation. He also introduced me to academics, grantwriting, and teaching, and made sure I had a position seven months before I graduated. I am deeply grateful for all the time he invested in me, and I resolve to do the same for any future students I work with.

My committee ensured that the research in this dissertation met the high standards of both BAE and CSE. With my co-chair, Dr. Deb, I had the privilege to work with a pioneer in evolutionary multi-objective optimization. He taught me so much about computer science education, on writing elegant algorithms, and how to make excellent graphs in gnuplot. Dr. Esfahanian ensured the algorithmic rigor and impact of this research. Finally, Dr. Harrigan kept the research grounded in the realities real farmers and encouraged me to look into the social dimension of optimization.

And of equal importance is my dear family. My partner Kaitlyn always encourages me to step out of my comfort zone, and I wouldn't have stepped foot in Michigan or Michigan State without her. She is equally a refuge on tough days and reminds me to enjoy life. And no one reminds me to stay curious more than my sweet Jonas, who everyday learns, observes, and practices. My parents John and Carol are the foundation of my love of school, learning, and teaching, and I'll always be grateful for all the opportunities that they provided me throughout my life. And I am also thankful to have a second set of parents, Laurie and Kevin Martinelli, whose love and support is fierce and endless. Last but far from least, this work would not exist without the hard work of its coauthors. Prakash Jha fields all of my thousands of agricultural questions, and no one is better to dream up new research than Sebastian Hernandez. Additionally, my many lab mates throughout the years have expanded my understand of the world, science, and friendship.

Michigan State University occupies the ancestral, traditional, and contemporary lands of the Anishinaabeg – Three Fires Confederacy of Ojibwe, Odawa, and Potawatomi peoples. In particular, the university resides on land ceded in the 1819 Treaty of Saginaw. We recognize Michigan's 12 federally recognized Native Nations, historic Indigenous communities in Michigan, Indigenous individuals and communities who live here now, and those who were forcibly removed from their homelands. In offering this land acknowledgement, we affirm Indigenous sovereignty, history, and experiences.

TABLE OF CONTENTS

LIST OF TABLES	X
LIST OF FIGURES	. xii
LIST OF ALGORITHMS	xvii
KEY TO ABBREVIATIONSx	viii
1. INTRODUCTION	1
 2. LITERATURE REVIEW	5 5 7 9 9 9 9 9 9 10 . 11 . 14
3 INTRODUCTION TO METHODOLOGY AND RESULTS	. 13
 4. BENEFITS OF SPARSE POPULATION SAMPLING IN MULTI-OBJECTIVE EVOLUTIONARY COMPUTING FOR LARGE-SCALE SPARSE OPTIMIZATION PROBLEMS 4.1 Introduction 4.2 Sparse population sampling 4.3 Experimental set up 4.3.1 Evaluating sparse population sampling 4.4 Results 4.4.1 Effects of SPS on standard EAs 4.4.2 Generic EA with SPS compared to SparseEA 4.5 Discussion 4.6 Conclusions 	. 19 . 19 . 22 . 25 . 31 . 31 . 35 . 40 . 43
 5. AGRICULTURAL INNOVIZATION: AN OPTIMIZATION-DRIVEN SOLUTION FOR SUSTAINABLE AGRICULTURAL INTENSIFICATION IN MICHIGAN	. 45 . 45 . 49 . 49 . 51

5.2.3 Experiment setup	
5.2.4 Prediction of developmental stages based on growing degree days	53
5.2.5 Common practice at the study site	
5.2.6 Innovization framework	
5.2.7 Validation	65
5.3 Results and discussions	66
5.3.1 Performance of irrigation recommendation strategy	66
5.3.2 Performance of nitrogen recommendation strategy	73
5.3.3 Performance of all recommendation strategy	74
5.3.4 Study implications	76
5.4 Conclusions	77
6. IMPROVED EVOLUTIONARY OPERATORS FOR SPARSE LARGE-SCALE N	AULTI-
OBJECTIVE OPTIMIZATION PROBLEMS	79
6.1 Introduction	79
6.2 Related Work	81
6.3 Proposed Method: Sparse NSGA-II	86
6.3.1 Varied Striped Sparse Population Sampling	87
6.3.2 Sparse Simulated Binary Crossover	
6.3.3 Sparse Polynomial Mutation	
6.3.4 Experimental setup	
6.4 Results	101
6.4.1 SMOP Results	101
6.4.2 Real-world problem results	105
6.4.3 Discussion	107
6.5 Conclusion	
7. CLIMATE-AWARE AGRICULTURAL MULTI-OBJECTIVE OPTIMIZATION	USING
STOCHASTIC WEATHER	110
7.1 Introduction	110
7.2 Methods	111
7.2.1 Stochastic weather generation	111
7.2.2 Stochastic EMO platform	
7.2.3 Preliminary Run Experimental Setup	114
7.3 Preliminary results	115
7.4 Conclusion	119
8. CONCLUSIONS	120
9. FUTURE RESEARCH RECOMMENDATIONS	121
APPENDIX	122
BIBLIOGRAPHY	152

LIST OF TABLES

Table 1: Complete set of hypotheses tested. Each cell includes 8 individual test configurations for each SMOP test function (denoted by SMOP[1–8]). Each configuration is then ran 60 times 3	or 0
Table 2: EFFECTIVE performance of NSGA-II, MOPSO, and MOEA/D-DE with varying number of decision variables. 3	er 3
Table 3: Comparative performance of NSGA-II with SPS and SparseEA. 3	7
Table 4. Optimization variables and constants 5	9
Table 5. Knowledge attributes 6	2
Table 6. Knowledge attributes recommendations	8

Table 11: Pareto optimal solutions for climate-aware optimization run, sorted by yield........... 118

Table A1. Effective performance of NSGA-II, MOPSO, and MOEA-D-DE with varying number of decision variables.	er 31
Table A2. p-value of two-sided Wilcoxon Rank-Sum test for the comparative runs	2
Table A3. Growing degree day in Fahrenheit during the innovization period for the vegetative an reproductive development stages 13	ıd 86

LIST OF FIGURES

Figure 6: Modeling overview	. 50
Figure 7. Study site	. 51
Figure 8. Flow chart of common irrigation practices in Cass County, Michigan	. 55
Figure 9. Flow chart of common nitrogen practices in Cass County, Michigan	. 57
Figure 10. The average gains and losses in yield between 1989 and 2018	. 67

Figure 11. Cumulative net increase in yield between irrigation recommendation strategy, nitrogen recommendation strategy, and all recommendation strategy strategies from common practice

Figure 21: 4-D Pareto front of average yields, average leaching, total irrigation, and crop failure risk. The bottom center sub-figure (a) is the Pareto set head-on, the top right sub-figure (b) is the Pareto set rotated to the right, and the top left sub-figure (c) is the Pareto set rotated to the left. 116

Figure A	A9. C	omple	te map	of all	validatio	n weather	stations	5	•••••	•••••	••••••	. 134

LIST OF ALGORITHMS

Algorithm 1: Sparse Population Sampling	
Algorithm 2: SSPS	91
Algorithm 3: S-SBX	96
Algorithm 4: S-PM	98

KEY TO ABBREVIATIONS

CPC	U.S. Climate Prediction Center
DAP	Days after planting
DSSAT	Decision Support System for Agrotechnology Transfer
DV	Decision variables
EA	Evolutionary algorithm
EMO	Evolutionary Multi-Objective Optimization
GCM	Global Climate Models
GDD	Growing degree day
GHCN	Global Historical Climatology Network
HV	Hyper-volume
IPCC	Intergovernmental Panel on Climate Change
LSMOP	Large scale multi-objective optimization
LSSMOP	Large sparse scale multi-objective optimization
МО	Multi-objective optimization
МОР	Multi-objective optimization problem
MRTN	Maximum Return to Nitrogen
NDFD	National Digital Forecast Database
NDS	Number of non-dominated solutions
NS	Non-dominated sorting
NSGA-II	Non-dominated sorting genetic algorithm II

NSGA-III	Non-dominated sorting genetic algorithm III
R	
S2S	Sub-seasonal to seasonal forecasts
SCF	Seasonal climate forecasts
SMOP	Sparse multi-objective problems
SPS	Sparse population sampling
V	Vegetative development

1. INTRODUCTION

Agriculture is a critical, and often overlooked, component of humanity's future well-being. Without enough food, humanity cannot exist in its current state. Therefore, a successful future depends upon our ability and willingness to maintain our environment, which provides food, water, and energy to humanity. In other words, our fate hinges on our ability to live sustainably.

However, the lifestyle of modern industrialized nations makes sustainable living increasingly difficult. The global population is expected to grow to 10 billion by the year 2050 (UN, 2019), and the existing population is becoming more affluent (Kharas and Gertz, 2010). Though increased affluence and population are positive in many ways, it makes it increasingly difficult for the earth to provide the food, water, and energy that we depend upon. First off, as we consume food, we consume and impact precious resources, like water, nutrients, healthy soil, and land. Therefore, as populations grow, so does humanity's demand for these necessary building blocks of food. Secondly, the food that more affluent communities consume, such as eggs, dairy, and meats, demands even more resources from the environment. And water quantity and quality are dwindling in many parts of the world (Awange, 2022), as is soil quality (Doran, 2002).

Our consumption of resources is not intrinsically unsustainable since many of these resources can be renewed. However, on the whole, today's food system is not renewable. We pump water to feed our crops without recharging the supplying aquifers. We apply so much fertilizer on fields that our waterways and seas are undrinkable and unfishable. We plow soil until all of its nutrients erode away. We destroy pollinator habitats to the point that we cannot pollinate our crops. And on top of all of these issues, climate change further endangers the viability and magnitude of crop production. Increased levels of greenhouse gases, such as carbon dioxide, methane, and

nitrous oxide will lead to farmland salination, aquifer depletion, increased weather uncertainty, decreased yields, and many more negative impacts (Alo et al., 2017). And industry is not the only source of greenhouse gases: agriculture accounts for 11% of greenhouse gases in the US (US EPA, 2020).

However, a sustainable future is not only concerned with the environment. Indeed, a successful and sustainable future includes social and economic goals as well (Purvis et al., 2019). For example, despite the importance of their work, farmers often lead economically precious lives, especially small and medium-scale farmers (Carolan, 2016). Furthermore, the average age of farmers is increasing (an average increase of 9.7 years from 1945 to 2017 (Johr, 2012)), implying that the next generation of workers is finding farming increasingly economically unsavory. Therefore, it is as important for farmers to follow practices that preserve their savings accounts as well as preserve the soil and water. From a social standpoint, modern farming practices often hollow out local farming communities in the Midwest (Lobao and Stofferahn, 2008). Agricultural mechanization turns many agricultural jobs obsolete, rendering once-thriving rural main streets into ghost towns. Unmitigated mechanization, therefore, makes many Midwestern communities unlivable for future farmers.

Though these goals of sustainability are plain and intuitive, it is deeply difficult to put them into practice. Management levers affecting each goal interplay with the other goals, creating a complex system of cause and effect. That is why modern scholarship takes a *systems approach*. For context, a traditional approach to agriculture has historically focused myopically on feeding people. Though this is important, it does not consider the entire agricultural ecosystem. These approaches often result in ecological disasters such as The Dust Bowl, where over-cultivation of the soil turned the fertile Great Plains into a barren landscape. A systems approach takes the entire

agro-ecological system at hand into account when making decisions (Lowrance et al., 1984). The Dust Bowl could have been prevented with alternative management practices like no-till farming or tree wind breaks while still providing food for our society (Rasmussen, 2010).

However, a challenge emerges when trying to sustain global agricultural systems: how to balance social, economic, and environmental sustainability? Unlike a 100-meter sprint, where there is always one clear goal and one clear winner, there are many equally good solutions to the problem at hand. For example, let us think of a hypothetical farmer musing over how to manage their plot of land. The farmer could take an entirely environmentally sustainable approach by rewilding their land and returning to a hunter-gather way of life. On the other hand, the best solution from an economic point of view would be to exploit the land to maximize profits. Finally, the best solution socially might be to invite all of her neighbors into an egalitarian Maoist farm commune. Each of these solutions is equally optimal since no single goal is more important than the other. However, each of these solutions would clearly be unpalatable to our farmer since they are terrible with respect to our other goals. Therefore, the task at hand becomes searching for the solutions that reasonably balance these goals. It turns out that there are innumerable solutions to agricultural management that are optimal concerning these three goals, known as a Pareto optimal set (Marler and Arora, 2004). As agricultural researchers following systems theory, our task should be to find the Pareto optimal solutions to sustainably support humanity.

As any farmer will tell you, the task of finding the perfect (i.e., Pareto optimal) solutions to agricultural management is practically impossible. There are innumerable ways of irrigating a field, applying nitrogen, choosing cover crops, and so on. And though one can plausibly come up with *good* solutions to a problem, finding *optimal* (i.e., perfect) solutions is remarkably challenging in agriculture. This is where computers become helpful. Computers excel at doing the grunt work

that humans struggle with, such as going through millions of possible management decisions and determining which are useful. Computers are particularly good at determining which solutions to a problem are Pareto optimal and which solutions are not. One management practice that would not be Pareto optimal, returning to our example of a farmer planning out their field, would be to throw salt on the field. This solution is clearly worse environmentally, economically, and socially than many other feasible solutions. The research field of finding Pareto optimal solutions among all possible solutions to a multi-objective problem is known as *multi-objective optimization* (Marler and Arora, 2004). With a Pareto set in hand, a farmer can do what we do best: make nuanced value judgments. Which solution, amongst the Pareto set, offers the best tradeoff between social, environmental, and economic goals? Alternatively, a farmer can examine the Pareto optimization: computers and humans work together to find the best solution to the problem at hand.

This research takes a multi-objective optimization approach to improve the sustainability of agriculture. Since the agricultural systems are so complex, existing multi-objective approaches are currently ineffective. Therefore, in our first and third studies, we developed novel multiobjective optimization methods that find increasingly better solutions to irrigation and nutrient management and run faster than existing methods. With these improvements, in our second study, we can create Pareto optimal sets for long periods to find common themes amongst management practices in a given region and climate. In our final study, we again use our improved optimization methods to improve crop management in the face of weather uncertainty predictively. With these strides in multi-objective and agricultural research, we have not only pushed the state-of-the-art in agricultural management, but in multi-objective algorithm design as well.

2. LITERATURE REVIEW

2.1 Optimization in Agriculture

2.1.1 Water and Nutrient management

In order to sustainably feed the growing population, we must first identify how much food will be needed and how to reach those targets. Estimates of our future food needs vary, but recent scholarship suggests that food production will need to increase by 70% by 2050 (van Dijk et al., 2021). To keep up with a 70% increase in food demand, today's agricultural systems need to dramatically change over the next 28 years. Since arable land, or land suitable for farming, is becoming increasingly rare (Thiaw, 2019), we must look to increase the output of our existing farmland to meet our target output sustainably. Fortunately, in many parts of the world, farms produce far less yield (i.e., food output) than their potential yield (Fischer et al., 2014). This difference in current and potential yields is called a yield gap, and closing yield gaps have been shown to be sufficient to meet the next generation's food demands (Pradhan et al., 2015). As an example, technologies like hybridization and improved management have increased average corn production in the US from roughly 25 bushels per acre to 140 bushels per acre (Jin et al., 2019; Wortmann et al., 2017). In other words, we can potentially meet our goals of sustainable agricultural production by optimizing how farmers take care of their crops.

However, which management practices do we optimize? At the most basic level, plants need CO₂, light, nutrients, and water to survive and thrive (Mauseth, 2014). Of these four requirements, conventional agriculture has the most control over water and nutrient management. Throughout the world, farmers irrigate their fields in order to provide plants with water during critical growth stages (Ritchie et al., 1986). Irrigation can supplement the water demands of plants at certain growth phases in temperate regions or it can fulfill nearly all of the crop's water requirements in arid or semi-arid areas. Nutrient management, on the other hand, provides crops with the critical elements needed for developing proteins and DNA. Water and air only provide hydrogen, oxygen, and carbon, whereas proteins and DNA also need phosphorus and nitrogen (Mauseth, 2014). Similar to irrigation, nutrient management provides these elements at the critical growth stages.

Improving nutrient and irrigation management can significantly improve yields. For example, in arid (i.e., dry) and semi-arid regions of the world, water demand often limits the growth potential of crops (Khan et al., 2006). However, by improving how farmers irrigate (e.g., delivery methods, amount, and timing) in arid regions, greater yields can often be achieved with the same amount of water (Parsinejad et al., 2013). Meanwhile, nutrient management aims to raise crops that are high yield, economically viable, and leave little environmental trace. Nutrient management includes its own unique challenges. Nutrients can be applied through organic means (e.g., compost, brown manures, green manures) or through industrial fertilizers (Robertson and Vitousek, 2009; Sharpley and Beegle, 2001). In regions where industrial fertilizers are scarce, management optimization can help make the most out of the scarce organic mediums, similar to optimization irrigation. On the other hand, where industrial fertilizers are available, farmers often have the means to over-apply nutrients (Robertson and Vitousek, 2009; Sharpley and Beegle, 2001). However, overabundance causes its own issues. Though the crops may have all the nutrients they need, the over-application of synthetic fertilizer can pollute surrounding waterways. The pollution from a single farm adds up when examined on a global scale. The cumulative pollution of waterways from thousands of farms leads to oceanic dead zones thousands of miles wide (Essick and Charles, 2013).

Though these two systems may seem independent, the best way to manage irrigation and nutrient uptake is as a single system. For example, nutrient uptake can be improved by well-timed irrigation applications (Kropp et al., 2019) and plants can use water more efficiently when their nutrient demands are met (Waraich et al., 2011). Furthermore, poor irrigation management can worsen the total amount of nitrogen polluted into the environment (Gheysari et al., 2009). Therefore, it makes sense to manage both irrigation and nitrogen application as a single, interconnected system.

2.1.2 Weather uncertainty and modeling

Irrigation and nutrient management can only go so far in the face of weather uncertainty. Farmers across the world experience dramatic differences in yield from year to year, mostly from yearly variations in weather (Coffel et al., 2022; Kukal and Irmak, 2018). This makes sense, considering the importance of water in plant growth and development. On top of that, climate change will increase weather uncertainty, including more intense extreme weather events like drought and heavy rain (Martinez-Feria et al., 2020). Therefore, it is essential to optimize plant growth holistically by incorporating nutrient management, irrigation management, and climate variability.

With this three-part system in mind, agronomists and farmers must determine which practices will effectively fulfill our environmental and nutritional needs. However, determining whether a given practice will be effective or not poses its own challenges. Historically, researchers have tested and evaluated different management methods through field experiments in the real world (Vilayvong et al., 2015). As one can imagine, this process takes a great deal of time since each management method requires a full growing season to complete. Furthermore, field tests are limited to the weather at the test site, making evaluating future outcomes of practices impossible.

Thankfully, crop simulation models solve many of the drawbacks of field tests. A crop simulation model is a computer program that takes in soil, weather, cultivar, and management inputs for a given season and simulates the performance of crops (e.g., yield and biomass) (Hoogenboom et al., 2017). This allows researchers to implement different management practices under any climate scenario imaginable. On top of that, modern computers can run most field-scale crop simulation models in fractions of seconds. With crop models, one can evaluate a management practice for thousands of plausible weather scenarios, thus ensuring that the given practice poses low risks to farmers, as discussed in the next section.

2.1.3 Stochastic weather generators

Due to the infeasibility of predicting complete and accurate seasonal forecasts, agronomists instead try to generate large numbers of *plausible* weather scenarios. With a set of plausible weather scenarios, agronomists can use crop simulation models to create and study a range of possible outcomes for their management decision. This is known as a Monte Carlo analysis (Ines and Mohanty, 2008). With this information, practitioners can understand the risk associated with a given decision. For example, imagine a farmer deciding between two management decisions. One management decision, decision A, leads to high average yields, but has a high risk of crop failure over a range of different weather scenarios. Next, imagine another management decision, decision, decision B, which leads to moderate yields with minimal risk of crop failure. Decision makers can now quantify not only the expected yields from a management decision but also the risk involved.

These sets of plausible weather scenarios can be generated through stochastic disaggregation. These take advantage of the fact that seasonal forecasts (e.g., average temperature in May, average rainfall in April) are very accurate with the advent of global climate models (GCMs) (Hansen et al., 2006). With these accurate aggregate forecasts, stochastic disaggregation

algorithms create random weather patterns that are statistically identical to the aggregate forecast (Han and Ines, 2017). Take this simplified scenario as an example: say, a GCM predicts there will be an average rainfall of 5mm within the next three days. One realization of this prediction would be 5 mm on day one, 0 mm on day two, and 10 mm on day three. Another realization would be 0 mm on day one, 5 mm on day two, and 10 mm on day three. Both of these scenarios have the same average rainfall and are therefore plausible outcomes for the next three days.

2.1.4 Crop Optimization

Despite the gains from crop simulation models and stochastic weather generators, the complexity of farming still makes finding global optimal management practices difficult. In other words, there are more possible ways for a farmer to manage their field than anyone could ever evaluate within a reasonable amount of time (even with a crop simulation model). Thankfully, computer science research has developed sophisticated methods for searching through these kinds of complex systems for optimal solutions. These methods, known collectively as optimization, have been proven to dramatically improve crop management practices (Eeswaran et al., 2021; Kropp et al., 2019, 2022b). The following sections dive into the challenges and opportunities in applying computer optimization to agricultural management.

2.2 Successes and Challenges in Multi-Objective Optimization in Agriculture

2.2.1 Introduction to optimization

Abstractly, optimization algorithms look for the best solution to a given problem (Goldberg, 1989). In order to do so, optimization platforms require an objective function that tells the optimization algorithm how good or bad a solution is to the problem at hand (known as a fitness value). The objective function takes, as input, the numerical solution to the problem, and outputs

the fitness of the given solution. This allows the optimization algorithm to search through all possible ways of solving a problem for the best possible approaches. In the case of agricultural optimization, a crop simulation model acts as an objective function by telling the optimization algorithm the fitness of a given management strategy. Additionally, optimization algorithms accept lower and upper bounds on the solution values (e.g., no negative irrigation values!), as well as constraints on what makes a solution feasible (e.g., no crop failure).

Traditionally, optimization algorithms handled relatively simple objective functions. These so-called classical optimization algorithms, such as golden section search, gradient descent search, or simplex search, typically could only handle objective functions that were unimodal, differentiable, and/or linear (Deb, 2009). More simply put, these algorithms solve specific classes of problems exceedingly well. Despite the importance of these problem sets, most real-world problems, such as crop optimization, are too complex to be solved by these classical methods. Real-world problems often have massive or difficult to navigate decision spaces, which classical methods struggle to or are incapable of solving.

2.2.2 Evolutionary Optimization

Evolutionary algorithms (EAs) were developed to solve the vast majority of optimization problems that are too complex to solve with the classic methods. Unlike classical methods, which perform extremely well in a small class of problems, EAs perform decently well for a vast class of problems (Goldberg, 1989). This makes EAs great methods for solving complex problems like agricultural management optimization (Kropp et al., 2019).

EAs, as their names suggest, mimic the process of biological evolution in order to find increasingly better solutions to a given problem. This generally works by first creating a random set of solutions to the given problem, and then virtually evolving the population into an incrementally fitter population (Goldberg, 1989). Under this evolution metaphor, each solution is an individual in a population, and the decision variables in the problem are the individual's DNA. Similar to biological evolution, the solutions that survive each generation of evolution depend on their fitness, which the objective function (i.e., fitness function) provides. After determining the survivors in a population, EAs then mate and recombine the best solutions to the problem in order to combine and carry on the best genes into the next population. Finally, a predetermined number of genes in the gene pool are mutated to introduce novel traits into the population. After many generations of evolution, EAs typically find near-optimal solutions, if not globally optimal solutions.

Randomness is an essential distinction between classical optimization methods and EAs. On the one hand, classical optimization algorithms are deterministic algorithms, or contain no randomness (Deb, 2009). In other words, if repeatedly ran, classical optimization algorithms will produce the exact same results over and over again. On the other hand, EAs are stochastic algorithms, or are driven by randomness. Theoretically, EAs will return different solutions to the optimization for every run. In the real world, where real random number generators do not exist, EAs often take a seed (i.e., any integer value) that will power the pseudo-random number generator that drives the evolutionary process.

2.2.3 Evolutionary Multi-Objective Optimization

So far, we have only discussed the complexity of problems in terms of how many input variables they contain. However, complexity can also arise when introducing new objective values. As a simplified example, imagine a person choosing one out of several houses in a housing market with two goals (i.e., objectives): price and square footage. Assuming all houses are of similar build quality, the objectives of price and square footage conflict with each other since bigger houses generally cost more. Therefore, there are at least two equally optimal solutions at hand: one that is large and expensive and one that is small but inexpensive. And between these two extreme points are houses that balance these two objectives: medium-sized houses with moderate prices. All of these solutions are known as Pareto optimal solutions and represent the two-dimensional objective space of the given problem (Marler and Arora, 2004). This demonstrates an important point: as the number of conflicting objectives climbs, so does the dimensionality of the results to the problem (Zille and Mostaghim, 2017). In other words, a single objective problem has one solution, a two objective problem has a two-dimensional solution set, a three objective problem has a three-dimensional solution set, and so on.

Though each solution in a Pareto set is fully optimal in a multi-objective problem, the chosen solution depends on the decision maker running the optimization. Going back to our home buyer example, a family (i.e., decision maker) on a tight budget might look at the Pareto set of houses and choose the cheaper house with less square footage. Another decision maker, say a single wealthy businessperson, might find the trade-off of increased price worth the extra square footage. The chosen solution depends entirely on the values of the decision maker, which highlights the power of multi-objective optimization: computers do the grunt work of finding the Pareto optimal set, and humans make the value-based decisions that computers cannot make.

Though there are classical means of solving multi-objective optimization problems (e.g., weighted sum, Benson's, and ε -constraint approaches), evolutionary multi-objective optimization (EMO) has experienced tremendous popularity over the last 30 years (Deb, 2009). EAs are excellent methods for solving multi-objective optimization problems, since the population of an EA is an effective way of representing Pareto optimal sets (Schaffer, 1985). However, ranking the

solutions in a population by fitness (a critical stage of all EAs), poses a challenge: how do you rank a population when there are several equally important objectives? The non-dominated sorting (NS) algorithm elegantly solves this problem (Deb et al., 2002).

In non-dominated sorting, the members of a population are grouped into increasingly better Pareto sets (non-dominated solutions is a synonym for Pareto optimal solutions), which ranks solutions by their fitness with respect to *all* objectives simultaneously. To do so, NS first finds the non-dominated solutions in a population, which are the solutions that are not dominated by any other member of the population. A solution A dominates solution B if and only if A is better than B in all possible objectives. Thus, the first non-dominated set of solution outperforms the other population members and are marked as the best solution in the population. Then, these best solutions are temporarily removed from the population, and the non-dominated set of the remaining population is found. This new set of non-dominated solutions is thus the second-best set of solutions in the population. This process continues until all population members are sorted into sets of solutions ranked with respect to all objectives.

However, what if an EA wants to select the top 50 solutions out of a population of 100 members, and the top two non-dominated sets have 25 and 30 individuals, respectively? Clearly, the algorithm would first select all members of the first non-dominated set, but how do we then select the best 25 members of the second non-dominated set? Different algorithms handle this problem differently, but an effective approach is to pick the best 25 solutions out of the 30-solution set with respect to diversity (Deb et al., 2002; Deb and Jain, 2014). In optimization, diversity refers to the similarity of solutions in a Pareto set. Generally, decision-makers are attracted to Pareto sets with more diversity, since it provides more options to choose from. Therefore, prioritizing solutions by diversity helps balance the sorting process, which has up to this point only prioritized

population optimality (also known as convergence). Some algorithms, specifically the nondominated sorting genetic algorithm-II (NSGA-II), determine diversity by calculating the volume between neighboring population members in a front (Deb et al., 2002). However, this method does not scale well when calculating the crowding distance between solutions in problems with more than three objectives. The algorithm NSGA-III overcomes this hurdle by creating many approximately-equally-spaced reference vectors in hyperspace to determine crowing (the inverse of diversity) (Deb and Jain, 2014). Non-dominated-sorting-based evolutionary algorithms are demonstrably among the top algorithms for solving multi-objective optimization problems, though there are still room for improvement in the field (Coello Coello et al., 2020).

2.2.4 Large-Scale and Sparse Optimization

Optimizing systems becomes increasingly difficult as the number of decision variables increases. Irrigation optimization often entails optimizing large numbers of decision variables, since each day of a, say, 100-day growing season, has a variable amount of irrigation that can be applied. As the number of decision variables increases, the number of possible solutions to a problem increases exponentially, which in turn renders the problem dramatically harder to solve. The literature classifies these types of problems, which have approximately 100 or more decision variables, as large-scale multi-objective optimization problems (LSMOP) (Zille and Mostaghim, 2017).

Researchers have developed several approaches to solve LSMOPs more effectively, but one approach works particularly well for irrigation optimization. This approach, known as sparse LSMOPs optimization, takes advantage of the sparsity of an optimal irrigation schedule (Tian et al., 2021b). Sparsity, in this context, refers to the ratio of zero to nonzero values in the irrigation schedule, or the number of days of no irrigation versus the number of days with irrigation. In many crops, such as corn, the number of days *without* irrigation far outnumber the number of days with irrigation (Kelley, 2016; Kelley and Hall, 2020). By taking advantage of this property, we can reduce the complexity of these otherwise challenging problems. Several optimization algorithms exist that solve sparse LSMOPs. However, they still suffer poor performance, and some run dramatically slower as the number of decision variables increases (Kropp et al., 2022a; Tan et al., 2021; Wang et al., 2021).

2.2.5 Innovization

As discussed earlier, weather uncertainty poses a practical challenge to agricultural management optimization for a future growing season. Crop simulation models, which are critical for most agricultural optimization platforms, are at the root of the issue. Crop simulation models require precise rain, solar radiation, minimum temperature, and maximum temperature data for the entire growing season in order to provide accurate predictions of yields (Hoogenboom et al., 2017). However, weather forecasts are less accurate after several days into the future. Therefore, agricultural optimization platforms need to rectify the need for accurate forecasts with the reality of inaccurate forecasts. Innovization is one method that addresses this issue.

Abstractly, an innovization platform finds innovations in a system by analyzing the output of an optimization run (Deb and Srinivasan, 2006). The output of a multi-objective optimization run, a Pareto set, is known to be good solutions to the problem at hand, and thus can be analyzed for what exactly makes them good solutions. In our proposed agricultural innovization, each multiobjective optimization scenario would be run against historical observed weather over 30 years, which would then be analyzed for recurring themes in the optimal solutions (Kropp et al., 2022b). Therefore, the benefit of innovization over classic optimization is that instead of running a separate optimization run for any weather scenario, innovization creates general improvements to an agricultural system that can be applied to any weather scenario.

3. INTRODUCTION TO METHODOLOGY AND RESULTS

All of the methodologies in this dissertation search for the best management practices for irrigation and nitrogen application for growing corn. Each of the individual studies addresses a specific challenge within this broad goal and is interconnected with the other studies. These challenges are pulled from two primary unaddressed issues from my master's thesis: accounting for weather uncertainty in optimization and speeding up the rate of optimization. In my master's thesis, excellent management practices were discovered using evolutionary multi-objective optimization, but weather was kept constant, and it took approximately two weeks to complete one optimization run. These two issues are interconnected since our hypothesized methods for optimizing under uncertain weather conditions (i.e., innovization or stochastic weather generation) dramatically increase run time, which makes speeding up the rate of optimization essential. In summary, our first and third studies addressed optimization speed, making the second and fourth studies on weather uncertainty feasible.

Our first study, *4*., seeks to improve the speed and convergence of evolutionary multiobjective optimization in solving irrigation-like problems. This class of problem, known as sparse large-scale multi-objective problems (LSMOP), includes problems with 100 or more decision variables, most of which are optimally zero values. Through a novel population initialization routine in the evolutionary optimization called sparse population sampling (SPS), we significantly improved the performance of three optimization algorithms (i.e., evolutionary, differential, and particle swarm optimization) in solving sparse LSMOPs.

In our second study, *5*., we use the methods of the first study to quickly find optimal agricultural practices in the face of weather uncertainty through innovization. Innovization worked by first running 900 corn management optimization runs over 30 years in a single site in Southwest
Michigan, and then data mining for common traits of optimal solutions. The results of the data mining are site-specific recommendations to improve the common practices in the study area that can be applied under any weather condition. The innovization platform used the SPS routine, which enabled it to run within a reasonable amount of time. Without SPS, 900 crop optimization runs, each taking 2 weeks to complete, would have taken 34 years to complete.

Though SPS improved the performance of multi-objective algorithms in solving sparse LSMOPs, the solutions were still far from perfect in solving many test problems. The third study of this dissertation, *6.*, aimed to create a better sparse LSMOP algorithm by creating advanced sparse operators for evolutionary optimization (i.e., population initialization, crossover, and mutation). In doing so, we will be able to solve climate-uncertain crop optimization more effectively through innovization and stochastic weather generation.

In our final study, we again use our improved sparse LSMOP algorithms in order to optimize crop management under weather uncertainty. Instead of using innovization, this study uses stochastic weather generators in order to quantify the risk within a given management decision.

4. BENEFITS OF SPARSE POPULATION SAMPLING IN MULTI-OBJECTIVE EVOLUTIONARY COMPUTING FOR LARGE-SCALE SPARSE OPTIMIZATION PROBLEMS

4.1 Introduction

At their core, most practical problem-solving tasks can be reduced to an optimization system. Every problem consists of goals (i.e., objectives) that we seek to improve by choosing among a set of plausible actions (i.e., decision variables). The decision variables that bring us closest to our goals are then chosen. Though we can deterministically find the exact optimal solution for certain problems (e.g., finding the minima or maxima of differentiable problems), a large class of problems are exceptionally difficult having either multimodality, nondifferentiability, or a large decision space. Classical point-based methods and other known optimization methods are thus not guaranteed to find the exact optimum (Deb, 2009). This broader class of difficult problems has inspired the creation of approximate yet flexible algorithms, such as evolutionary algorithms (EA), to attempt to find a near-optimal solution in a quick computational time. EAs harness the concept of natural selection and genetics to generate increasingly better solutions to difficult optimization problems by exploiting an evolving population of solutions. Starting with a population of random solutions to a given problem, EAs mimic evolution by evolving this population into incrementally improved solutions over generations of different populations. Furthermore, the genes of each population member are the decision variables of the problem, and the survivability of each population member is how well it solves the problem, or objective value (Goldberg, 1989).

EAs are particularly effective at solving multi-objective optimization problems (MOP) (Deb, 2009; Storn and Price, 1997; Zitzler et al., 2001). MOPs consist of multiple objectives that

are in conflict with each other (Tamaki et al., 1996), and are endemic to critical fields of engineering, such as hydrological engineering (Herman et al., 2020; Hernandez-Suarez et al., 2018), agricultural engineering, (Kropp et al., 2019; Whittaker et al., 2017) mechanical engineering (D'Errico et al., 2011), and structural engineering (Suresh et al., 2007). Since, unlike single-objective problems, MOPs have multiple objectives, their solutions can be expressed in the form of a set of solutions, or a Pareto set (Deb, 2009). For example, there is often multiple equally optimal solutions when training a machine learning model: one solution might have a low training error with high over fitting, and another might have low over fitting at the expense of training accuracy. Evolutionary multi-objective (EMO) algorithms combine the principles of EAs into finding Pareto-optimal solutions to challenging MOPs (Coello, 2006).

Amongst MOPs, there is a subset that has received little stand-alone attention in the literature: large-scale sparse MOPs (LSSMOP). LSSMOPs are problems with high numbers of decision variables (Zille and Mostaghim, 2017), and sparse MOPs are problems whose optima are known to be sparse vectors (Gong et al., 2015). Sparse optimization is present in countless real-world scenarios. Portfolio optimization is one example (Eftekharian et al., 2017). In portfolio optimization, an investor attempts to optimally distribute funds into a large number of available investments. However, portfolios are often sparse solutions, since most investors can only manage and desire a relatively small number of investments (Branke et al., 2009). When considering thousands of possible investment scenarios, the problem immediately becomes massive. Another example is agricultural irrigation optimization. Farmers in semi-arid and arid regions often require irrigation applications. Therefore, farmers must decide what minority of days of the growing season to water their crops, requiring a sparse optimal solution (Kropp et al., 2019). Multi-

objective optimization is also widely found in feature selection, where an optimal subset among many features must be chosen, minimizing the number of features chosen and maximizing model accuracy (Zhang et al., 2017). Outside of these specified cases, there are few general-purpose EMOs developed for LSSMOPs with evolutionary optimization, besides the recent algorithm SparseEA (Hong et al., 2021; Ye Tian et al., 2020).

Population sampling strategies have been under-studied in the LSSMOP literature, despite the fact that uniform genome sampling employed by most EAs is not well suited for LSSMOPS. For example, real-valued EAs have genomes consisting of real numbers (Herrera et al., 1998). They typically initialize their population by pulling random real values from a uniform distribution for each location of the genome (Whitley, 1994). Though this methodology effectively solves problems that converge upon dense genomes in low-dimensional decision spaces, it struggles to overcome problems that are both large-scale and sparse (Ye Tian et al., 2020). This is likely because solving large-scale sparse MOPs with uniform sampling requires evolving the whole initial population from a dense high-dimensional to a sparse high-dimensional space. Take the following theoretical problem as a demonstrative example. Say, $F(\mathbf{x})$ is an optimization problem, where $\mathbf{x} \in \mathbb{R}^{1000}$, and 95% elements of \mathbf{x}^* are zero. Firstly, it is deeply unlikely for an initial population created by uniform distribution within the lower and upper bound of \mathbf{x} to be sparse. If we are using 32-bit real values and uniform genome sampling, then there will be a $\frac{1}{4.294.967.296}$ chance that a single member of the genome will be set to zero, let alone the 949 other genes that must be set to zero in \mathbf{x}^* . An EA, given this initial population, needs to then evolve each of the 950 genes to zero. Under these circumstances, there is room for significant speed and efficiency improvements for traditional initial population sampling.

The goal of this paper is to introduce a novel approach to solve LSSMOPs through sparse population sampling (SPS). This formerly un-researched approach is shown to significantly improve the performance of evolutionary optimization problems operating in sparse optimization spaces with a simple, linear time run time. Existing algorithms can easily incorporate SPS, which allows them to easily harness the benefits of this routine with minimal coding effort.

In order to examine this goal, we put forward two hypotheses: 1) algorithms that are not designed to solve LSSMOPs will solve LSSMOPS significantly more effectively when using SPS, and 2) the algorithms in Part 1, combined with SPS, will perform significantly better than algorithms designed to specifically to solve LSSMOPs.

In the following sections, the SPS concept is described in Section 4.2 Sparse population sampling. The experimental set-ups are then discussed in Section 4.3 Experimental set up. Extensive results to evaluate the SPS concept are presented in Section 4.4 Results. Discussions of the results and conclusions of this study are presented in Sections 4.5 Discussion and 4.6 Conclusions, respectively.

4.2 Sparse population sampling

Instead of sampling a population uniformly, which statistically always leads to a majority of non-zero values, one can simply seed the population of an EA with sparse initial genomes. This provides the population with a head start towards reaching the optima known to have a sparse solution. Sampling a population with sparse individuals is the core concept behind our proposed algorithm, SPS. SPS seeds a population with individuals within a range of sparsities, or with a single sparsity when expert knowledge is available. The SPS routine that we propose is as simple as it is effective. Assume the given problem has *d* decision variables and is a population-based algorithm with *N* population members. SPS works by first generating a population with an existing population sampler (e.g., a uniform sampling distribution) for the given algorithm (Algorithm 1, line 4), and then modifying this population to ensure all population members are within certain sparsity limits (Algorithm 1 lines 5–15). These limits will be known as b_l for the lower bound, b_u for the upper bound, with $b_l, b_u \in$ [0,1], where 1 represents 100% sparsity (i.e., a genome of all zeros), and 0 represents 0% sparsity (i.e., a genome of all non-zero values). If a target sparsity is known through expert knowledge, *b l* and *b u* can both be set to that target value. This target sparsity range will be referred to as *S*, such that $S = [b_l, b_u]$. If no range or target sparsity is given, SPS will seed the population between a sparsity of 50% and 100%, which is likely to cover the majority of target sparsities of sparse problems (Algorithm 1, lines 1–3).

With the sparsity range *S* initialized, SPS then creates one unique sparsity mask (*M*) for each of the *N* population members (Algorithm 1, lines 6–14). An individual's sparsity mask is an array that contains a corresponding bit for all *d* elements of the genome for the given problem. If the bit is zero, that location in the genome is forced to be a zero in the initial population. Which bits are one and which bits are zero are generated at random (Algorithm 1, lines 6–14) and differ between individuals in the initial population. The sparsity of the mask *M* lies within the following given range:

$$\left\{ M|1 - \frac{\sum_{i=0}^{d} m_i}{d} \in S \right\} \tag{(1)}$$

where *M* is the sparsity mask, m_i is the value of the mask at position $i \ (m \in \{0,1\}), d$ is the number

Input: N, d, initPop, b_l , b_{μ} // $N \rightarrow$ population size // $d \rightarrow$ # of decision variables or dimensions // initPop \rightarrow parent algorithm's population sampler // $b_l \rightarrow$ Lower sparsity bound // $b_u \rightarrow$ Upper sparsity bound Output: P // P \rightarrow (the initial population) 1 if b_l and b_u are undefined then $b_l \leftarrow 0.5;$ 2 $b_u \leftarrow 1;$ 4 $pop \leftarrow initPop(N, d);$ // Make sparsity mask for each individual 5 for indiv $\leftarrow 1$ to N do // Calculate number of zeros to enforce minZeros $\leftarrow d \times b_l$; 6 7 maxZeros $\leftarrow d \times b_u$; $zCount \leftarrow randint(minZeros, maxZeros);$ 8 // Create sparsity mask (M) $M \leftarrow array of d ones;$ 9 /* Zero out enough random positions in the mask (M)to enforce target sparsity for this individual */ for $z \leftarrow 1$ to zCount do 10 11 $random \leftarrow randint(0, d);$ M[random] = 0;12 end 13 // Bit-wise multiplication to enforce sparsity $pop[indiv] = pop[indiv] \times M$ 14 15 end 16 return pop

Algorithm 1: Sparse Population Sampling

of dimensions in the genome, and *S* is the target sparsity range ($S = [b_l, b_u]$). This relationship ensures that every member of the population is rendered within the target sparsity range. The sparsity mask for each individual will also be unique, in order to encourage a good spread of sparsities in the initial population. It is important to note that the position of the non-zero and zero values is not maintained after the initial population generation. SPS is an efficient and adaptive addition to any existing algorithm. It only runs once during the population initialization with a linear run time with respect to population size. It is also flexible, since it can be incorporated into any population-based algorithm, including standard EAs.

4.3 Experimental set up

4.3.1 Evaluating sparse population sampling

When measuring the performance of SPS, there are a myriad of independent performance parameters that affect the dependent performance indicators of optimization runs. The independent parameters include:

- 1. use of SPS
- 2. choice of test function
- 3. number of decision variables 4. sparsity while the dependent

and performance indicators include:

- 1. hyper-volumes (HV)
- 2. run time (R)
- 3. number of non-dominated solutions (NDS)

This section describes the battery of hypotheses and their experiments that measure the sensitivities of independent parameters on the dependent performance indicators.

The two main hypotheses stated at the end of the introduction will now be called *effective* hypotheses and *comparative* hypotheses. The goal of the *effective* hypotheses is to understand the effects of SPS on evolutionary and population-based algorithms, and its sensitivity to sparsity. This is achieved by running standard EAs under two configurations: one with SPS and another

without SPS. The goal of the *comparative* runs is to determine whether an existing algorithm using SPS is sufficiently effective to act as a stand-alone sparse optimization algorithm. This is achieved by measuring the performance of a standard EA with SPS against stand-alone sparse EA algorithms. As of today, the primary generic sparse optimization EA is SparseEA (Ye Tian et al., 2020). The standard EA used in this case was the Non-dominated Sorting Genetic Algorithm (NSGA-II), which performed the best with SPS in the *effective* results. These overarching hypotheses could not be tested using a single test since there are so many independent parameters and performance indicators. Therefore, they are broken into smaller sub-tests that account for these differences (Figure 1).

Each of these sub-tests utilized statistical significance testing to verify its particular research question. The first step in statistical significance testing is to determine the null hypotheses, which represents the status quo. Preliminary runs showed that generic EAs have very low HVs when solving LSSMOPs, a metric of solution diversity and convergence (While et al., 2006). Therefore, our first null hypothesis for the *effective* runs is that SPS does not significantly improve mean HV (H_0^{eHV} , Figure 1, Table 1), and our alternative hypothesis asserts that SPS significantly increases the mean HV in LSSMOPs. Also, during preliminary runs, generic EAs often had low numbers of non-dominated solutions (NDS) in their final populations. This is an unsuitable outcome, since having more solutions in a resulting Pareto front gives decision-makers more optimal solutions to choose from. Since there is a hard upper limit to NDS (the population of the EA), and some EAs already retain 100% of their NDS, our second null hypothesis is that SPS will worsen the NDS in final populations of LSSMOPs (Figure 1, Table 1), and our alternative hypothesis is that SPS will maintain or improve NDS in LSSMOPs. Run time is not considered

for the *effective* runs because run time will not be significantly affected by the use of SPS, since it runs in linear time with regards to population size, which is computationally negligible.

For the *comparative* runs, the first null hypotheses assert that NSGAII with SPS does not significantly improve the HV (H_0^{cHV}) and run time (H_0^{cR}) compared to SparseEA (Figure 1, Table 1) for LSSMOPs. Our alternative hypothesis is that NSGA-II combined with SPS will perform significantly better with respect to HV and run time than SparseEA. The final null hypothesis is that the NDS under NSGA-II with SPS will be worse than with SparseEA (H_0^{cN} , Figure 1, Table 1), and the alternative hypothesis is that NSGA-II will maintain or improve upon SparseEA's NDS.

Each of the above hypotheses (H_0^{eN} , H_0^{eHV} , H_0^{cN} , H_0^{cHV} , H_0^{cR}) were then further split into sub-hypotheses to test and measure the effect of algorithm complexity (i.e., number of decision variables) and different test problems (Figure 1, Table 1). The number of decision variables and algorithm were generated by the Sparse Multi-Objective Problems (SMOP) test suite (Ye Tian et al., 2020). SMOP consists of 8 test problems (SMOP1 through SMOP8), each designed with a different optimization challenge (e.g., multimodality, deception) and Pareto front shape. Furthermore, each of these test functions can contain different sparsities and decision variables and a target sparsity to 90%, or 400 decision variables and a sparsity of 85%. The flexibility of SMOP allows us to measure the effects of problem characteristics, such as sparsity and number of decision variables, on our overall hypotheses.



Figure 1. Hierarchy of hypotheses. a.) and b.) are effective (H_0^e) and comparative (H_0^c) hypotheses respectively, which are then broken down into performance metrics, and then different run constants. The performance metrics include number of non-dominated Solutions (NDS) $(H_0^{eN} \text{ and } H_0^{cN})$, hyper-volume (HV) $(H_0^{eHV}, \text{ and } H_0^{cHV})$, or run time (H_0^{cR}) . The run constants include test problem from the Sparse Multi-Objective Problems (SMOP), and number of decision variables (DV). c.) breaks down hypotheses by number of decision variables, and is denoted as H_0^{qK} : *U*/SMOP [i-j], where, *q* is either *e* for effective or *c* for comparative , *K* is either *HV*, *N*, or *R* for HV, NDS, or run time respectively, *U* is the number of decision variables, and SMOP [i-j] denote a range of SMOP test functions, such as SMOP[1–8] denoting SMOP1 through SMOP8.

The exact test function configuration (i.e., complexity, sparsity, and test problem) generated for each hypothesis differed between the *effective* and *comparative* runs. For the *comparative* runs, we wanted to measure how NSGA-II and SparseEA both performed against

each other under increasing algorithmic complexity. To achieve these ends, the *comparative* runs hold sparsity at a typical constant value (90%) while varying the number of decision variables between 100, 500, 1,000, 2,500, 5,000, and 7,500¹. To measure these effects, the main *comparative* hypotheses (H_0^{cN} , H_0^{cHV} , H_0^{cR}) were divided up into individual hypotheses and will be denoted as H_0^c : *U/SMOPL*, where *U* is the number of decision variables in that given test, and *L* is the test problem used (e.g., H_0^c : 1000/*SMOP*4, H_0^N : 5000/*SMOP*5). The *effective* runs perform these same hypothesis tests, which vary the number of decision variables (while using the same notations), while also including a set of runs to observe the effect of varying sparsity on SPS. The goal of this subset of the objective runs is to observe the effect of sparsity on problems with and without SPS and will not be experiment based.

Each combination of problem type, sparsity, and decision variables was rerun several times. Based on preliminary simulations, we chose 60 runs to maximize the accuracy of our statistical measures while minimizing run time of all simulations. For every configuration, the HV and the number of non-dominated solutions are measured for each Pareto front. In preliminary runs, the reference point was (1,1). However, due to the large scale of many of the test problems, several algorithms could not find solutions within the square of (1,1) (see example in Figure A8).² This resulted in HV of zero for many trials, which prevents any meaningful analysis in higher decision spaces. On the other hand, reference points that are too far away from the Pareto front are imprecise. Therefore, in order to strike a balance, the smallest reference point that yielded all non-

¹ These intervals were chosen to first measure the finer effects of changes on the number of decision variables, and then the asymptotic behavior as the number of decision variables increased.

² Figures marked with a prefix 'A' are available in the Appendix.

zero HVs was chosen for the *comparative* and *effective* runs. Specifically, the *comparative* reference point is (2,2), and the *effective* reference point is (4,4). Since all test problems had extreme points between (0,1) and (1,0), no objective normalization was run. Optimization run time is also recorded for each run. Typically, the t-test is used to reject or not reject a null hypothesis and requires that data follows a consistent parametric distribution. However, none of the resulting metric distributions followed a single consistent parametric distribution, so we instead used a non-parametric hypothesis test: the Wilcoxon Rank-Sum test (Hollander et al., 2013). The Wilcoxon Rank-Sum test is a two-sided hypothesis test based on the relative ranking of the two samples in question (e.g., one with SPS and one without SPS). The p-value threshold used was 2.5%.

Table 1: Complete set of hypotheses tested. Each cell includes 8 individual test configurations for each SMOP test function (denoted by SMOP[1–8]). Each configuration is then ran 60 times.

Decision variables *	Global hypotheses						
	H_0^{eHV}	H_0^{eN}	H_0^{cHV}	H_0^{cN}	H_0^{CRT}		
500	H_0^{eHV} :500/SMOP[1-8]	H_0^{eN} :500/SMOP[1-8]	H ₀ ^{CHV} :500/SMOP[1-8]	H ₀ ^{cN} :500/SMOP[1-8]	H_0^{cRT} :500/SMOP[1–8]		
1,000	H_0^{eHV} :1000/SMOP[1-8]	H_0^{eN} :500/SMOP[1–8]	H ₀ ^{cHV} :1000/SMOP[1-8]	H ₀ ^{cN} :1000/SMOP[1-8]	H ₀ ^{cRT} :1000/SMOP[1-8]		
2,500	H_0^{eHV} :2500/SMOP[1-8]	H ₀ ^{eN} :500/SMOP[1-8]	H ₀ ^{cHV} :2500/SMOP[1-8]	H ₀ ^{cN} :2500/SMOP[1-8]	H_0^{cRT} :2500/SMOP[1-8]		
5,000	H_0^{eHV} :5000/SMOP[1-8]	H_0^{eN} :500/SMOP[1–8]	H ₀ ^{cHV} :5000/SMOP[1-8]	H ₀ ^{cN} :5000/SMOP[1-8]	H_0^{cRT} :5000/SMOP[1-8]		
7,500	H ₀ ^{eHV} :7500/SMOP[1-8]	H ₀ ^{eN} :500/SMOP[1-8]	H ₀ ^{CHV} :7500/SMOP[1-8]	H ₀ ^{cN} :7500/SMOP[1-8]	H ₀ ^{cRT} :7500/SMOP[1-8]		

^{*}All runs with varying decision variables keep sparsity at 90%. Each hypothesis has a label that corresponds to its configuration, denoted as H_0^{qK} : *U*/SMOP*L*, where, *q* is either *e* for *effective* or *c* for *comparative*, *K* is either *HV*, *N*, or *R* for HV, NDS, or run time respectively, *U* is the number of decision variables, and SMOP*L* is one of the 8 SMOP test functions. Finally, let [i-j] denote a range of values, such as SMOP[1–8] denoting SMOP1 through SMOP8.

For the *effective* runs, we used three commonly-used population-based multi-objective optimization algorithms: differential evolution (Storn and Price, 1997) (MOEA/D-DE (Zhang et al., 2009)), Particle Swarm optimization (Kennedy and Eberhart, 1995) (MOPSO (Lin et al., 2015)), and evolutionary optimization (NSGA-II (Deb et al., 2002)). For the *comparative* run, we ran SparseEA (Y. Tian et al., 2020a) (currently the sole generic algorithm designed to solve LSSMOPs) against the best performing algorithm with SPS from the *effective* runs (NSGA-II). All the *effective* and *comparative* optimization runs were tested using the MATLAB (version

2020b) (MATLAB, 2020) optimization library PlatEMO (Tian et al., 2017), and each algorithm ran under the default arguments of PlatEMO version 2.9.0. For the runs with SPS, the sparsity range was set to the default for SPS 50% to 100%. The computer was run on 2, 12-core 2.50GHz Intel Xeon processors, with 125GB of RAM.

4.4 Results

4.4.1 Effects of SPS on standard EAs

Across all algorithms, test problems, and numbers of decision variables in the *effective* runs, SPS significantly increased the HV of their respective algorithms (Table 2 and Figure A1). In other words, we rejected H_0^{eHV} :[100–7500]/SMOP[1–8] for NSGA-II, MOPSO, and MOEA/D-DE (save one exception H_0^{eHV} :100/SMOP6 for NSGA-II, which cannot be rejected). The increase in HV was more dramatic in some test problems, especially SMOP1, SMOP2, SMOP4, SMOP5, and SMOP8, all with p-values of 3.557e-21. (Figure 2, Figure A1b, Figure A1d, Figure A1e, Figure A1h, and Table A1). This rate of improvement increases with the number of decision variables (Figure 2 Figure A1). Lastly, relative to the algorithms without SPS, the algorithms using SPS kept a noticeably steady HV throughout different numbers of decision variables (Figure 2 and Figure A1) up to 7500 decision variables.

The effect of SPS on NDS was more varied during the *effective* runs. NSGA-II on average maintained its entire population of non-dominated solutions both with and without SPS (Table 2 and Table A1). This is a positive outcome, since NSGA-II was already able to achieve near perfect NDS without SPS, so any improvement was impossible. SPS even modestly improved the NDS for NSGA-II for SMOP8 under 500 and 1000 decision variables (Table 2 and Table A1). Therefore, H_0^{eN} :[100–7500]/SMOP[1–8] was rejected for NSGA-II. This significant improvement

of HV, while avoiding any manner of negative performance impacts (i.e., run time or NDS) across all test problems and numbers of decision variables, demonstrates the utility of using SPS with NSGA-II under LSSMOPs. The results of MOPSO with SPS were more mixed. MOPSO without SPS already exhibited a perfect median NDS across all test problems and numbers of decision variables, and the majority of MOPSO with SPS maintained these perfect NDS. Specifically, H_0^{eN} :SMOP [1 - 2], H_0^{eN} :[100-7,500]/SMOP4, and H_0^{eN} :[100-7500]/SMOP[7-8] were all rejected for MOPSO (Table 2 and Table A1). However, MOPSO with SPS had significant decreases in NDS under SMOP5 and SMOP6. With SPS, MOPSO had medians ranging between 28 to 30, whereas MOPSO without SPS had NDS of 100 (Table 2 and Table A1; Figures Figure A2e, and Figure A2f). Therefore, we were unable to reject H SMOP [5-6] for MOPSO. SMOP3, also posed a challenge to MOPSO with SPS, but the decreases were less dramatic, with p-values being 19 to 22 orders to magnitude smaller than the latter (Table A1). Also, 95% of values stayed between 80 and 95 NDS (Figure Figure A2c). Regardless, we were unable to reject H_0^{eN} :[100– 7500]/SMOP3. Yet, from a practical standpoint, though these runs failed the Wilcoxon Rank-Sum test, they are still decent solutions. Unlike the runs under SMOP5 and SMOP6, the majority of NDS deviations still included the vast majority of the population. This practically minor decrease is arguably a good tradeoff for the significant improvement in HV. MOEA/D-DE also had mixed results for NDS. Firstly, using SPS with MOEA/D-DE tended to increase NDS variability (Figure Figure A2). Secondly, we were only able to reject a minority (7 out of 48) of NDS null hypotheses for MOEA/D-DE (Table 2 and Table A1). However, similar to MOPSO-SPS and SMOP3, the significant decreases were not the majority of the population. Most deviations were above 90 NDS, and a relative few medians went down to 65 (Table 2, Table A1).

# dec.	Test	NSGA-II w/ and w/out SPS MOPSO w/ and w/out		nd w/out SPS	MOEA/D-DE w/ and w/out SPS		
vars.	prob.	Median HV	Median NDS	Median HV	Median NDS	Median HV	Median NDS
100	SMOP1	15.49(15.40)+	$100(100)^{\approx}$	15.31(14.36)+	100(100) [≈]	15.30(14.69)+	$100(100)^{\approx}$
100	SMOP2	15.47(14.60)+	100(100)≈	15.08(9.62)+	100(100) [≈]	$15.14(11.13)^{+}$	$100(100)^{\approx}$
100	SMOP3	15.47(13.92)+	100(100)≈	12.53(9.73)+	100(100)-	15.19(11.00)+	99(100)-
100	SMOP4	15.78(15.60)+	100(100)≈	15.77(13.58)+	100(100)≈	15.73(14.04)+	100(100)≈
100	SMOP5	15.77(15.18)+	100(100)≈	15.60(14.99)+	30(100)-	15.63(15.04)+	98(100)-
100	SMOP6	15.75(15.75)	100(100)≈	15.63(15.52)+	28(100)-	15.61(15.52)+	98(100)
100	SMOP7	15.19(14.82)+	100(100)≈	14.78(14.09)+	100(100)≈	14.92(14.44)+	94(85)+
100	SMOP8	$14.71(11.81)^{+}$	100(100)≈	14.03(4.90)+	100(100)≈	14.40(7.85)+	94(84)+
500	SMOP1	15.42(14.93)+	100(100)≈	15.32(14.07)+	100(100)≈	15.18(14.07)+	100(100)≈
500	SMOP2	15.30(13.00)+	$100(100)^{\approx}$	15.07(8.53)+	100(100)≈	14.96(8.79)+	$100(100)^{-}$
500	SMOP3	15.29(11.94)+	100(100)≈	12.37(9.18)+	$100(100)^{-}$	14.94(10.25)+	90(100)-
500	SMOP4	15.78(14.87)+	100(100)≈	15.77(13.15)+	100(100)≈	15.71(13.26)+	100(100)-
500	SMOP5	15.75(15.06)+	100(100)≈	15.62(14.95)+	28(100)-	15.59(14.97)+	97(100)-
500	SMOP6	15.72(15.64)+	100(100)≈	15.61(15.46)+	28(100)-	15.61(15.44)+	98(100)-
500	SMOP7	14.96(13.83)+	100(100)≈	14.75(13.85)+	100(100)≈	14.66(13.84)+	99(100)-
500	SMOP8	14.28(7.75)+	100(98)+	13.97(3.76)+	100(100)≈	13.84(2.87)+	98(94) [≈]
1,000	SMOP1	15.40(14.58)+	100(100)≈	15.31(14.02)+	100(100)≈	15.13(13.99)+	100(100)≈
1,000	SMOP2	15.26(11.78)+	100(100)≈	15.09(8.31)+	100(100)≈	14.96(8.55)+	100(100)-
1,000	SMOP3	15.15(10.64)+	100(100)≈	12.33(9.09)+	100(100)-	14.62(10.09)+	83(100)-
1,000	SMOP4	15.78(14.35)+	100(100)≈	15.77(13.06)+	100(100) [≈]	15.71(13.18)+	$100(100)^{-}$
1,000	SMOP5	15.75(14.93)+	100(100)≈	15.60(14.94)+	30(100)-	15.55(14.95)+	98(100)-
1,000	SMOP6	15.72(15.55)+	100(100)≈	15.61(15.45)+	29(100)-	15.63(15.43)+	98(100)-
1,000	SMOP7	14.88(13.04)+	100(100)≈	14.75(13.83)+	$100(100)^{\approx}$	14.65(13.71)+	99(100)-
1,000	SMOP8	14.23(4.75)+	$100(100)^{+}$	13.99(3.52)+	100(100)≈	13.95(2.56)+	97(99)-
2,500	SMOP1	15.38(13.89)+	$100(100)^{\approx}$	15.29(13.97)+	100(100)≈	15.14(13.92)+	100(100)-
2,500	SMOP2	15.24(10.18)+	$100(100)^{\approx}$	15.06(8.18)+	100(100)≈	14.89(8.44)+	100(100)-
2,500	SMOP3	15.19(8.63)+	100(100)≈	12.20(9.06)+	100(100)-	14.48(9.95)+	82(100)-
2,500	SMOP4	15.78(13.80)+	$100(100)^{\approx}$	15.77(13.01)+	100(100)≈	15.73(13.15)+	100(100)-
2,500	SMOP5	15.75(14.62)+	$100(100)^{\approx}$	15.61(14.94)+	29(100)-	15.59(14.94)+	99(100)-
2,500	SMOP6	15.72(15.40)+	$100(100)^{\approx}$	15.62(15.44)+	29(100)-	15.61(15.43)+	98(100)-
2,500	SMOP7	$14.83(11.73)^{+}$	100(100)≈	14.74(13.80)+	$100(100)^{\approx}$	14.55(13.64)+	94(100)-
2,500	SMOP8	14.19(2.12)+	100(100)≈	14.00(3.51)+	100(100) [≈]	13.90(2.25)+	95(100)-
5,000	SMOP1	15.38(13.24)+	$100(100)^{\approx}$	15.27(13.99)+	100(100)≈	15.12(13.90)+	100(100)-
5,000	SMOP2	15.23(9.22)+	$100(100)^{\approx}$	15.07(8.16)+	100(100)≈	14.90(8.40)+	100(100)-
5,000	SMOP3	15.17(7.40)+	100(100)≈	12.32(9.14)+	$100(100)^{-}$	14.01(9.90)+	73(100)-
5,000	SMOP4	15.78(13.49)+	100(100)≈	15.77(12.98)+	$100(100)^{\approx}$	15.73(13.14)+	$100(100)^{-}$
5,000	SMOP5	15.75(14.26)+	$100(100)^{\approx}$	15.60(14.94)+	27(100)-	15.55(14.94)+	98(100)-
5,000	SMOP6	$15.71(15.27)^{+}$	$100(100)^{\approx}$	15.62(15.44)+	29(100)-	$15.61(15.42)^{+}$	99(100)-
5,000	SMOP7	$14.81(10.34)^{+}$	$100(100)^{\approx}$	14.73(13.78)+	$100(100)^{\approx}$	14.35(13.58)+	91(100)-
5,000	SMOP8	$14.19(1.45)^{+}$	$100(100)^{\approx}$	$14.00(3.69)^{+}$	$100(100)^{\approx}$	13.37(2.20)+	63(100)-
7,500	SMOP1	15.38(12.80)+	$100(100)^{\approx}$	15.31(13.97)+	100(100)≈	15.10(13.90)+	100(100) [≈]
7,500	SMOP2	15.23(8.80)+	$100(100)^{\approx}$	15.07(8.10)+	100(100) [≈]	14.93(8.36)+	100(100)-
7,500	SMOP3	15.17(6.90)+	$100(100)^{\approx}$	12.17(9.00)+	$100(100)^{-}$	12.86(9.88)+	69(100)-
7,500	SMOP4	15.78(13.36)+	$100(100)^{\approx}$	15.77(12.98)+	100(100)≈	15.74(13.16)+	100(100)-
7,500	SMOP5	15.75(14.02)+	$100(100)^{\approx}$	15.60(14.94)+	28(100)-	15.53(14.94)+	98(100)-
7,500	SMOP6	15.72(15.19)+	$100(100)^{\approx}$	15.61(15.44)+	29(100)-	15.63(15.43)+	99(100)-
7,500	SMOP7	14.81(9.42)+	$100(100)^{\approx}$	14.74(13.80)+	100(100)≈	14.47(13.55)+	91(100)-
7,500	SMOP8	14.18(1.26)+	100(100) [≈]	14.03(3.49)+	100(100)≈	$13.44(2.15)^{+}$	66(100)-
Let ≈. a	et - x, and + denote statistical differences between the runs with and without SPS according to the Wilcoxon Rank-Sum test - denotes						

number of decision variables.

Let -, \approx , and + denote statistical differences between the runs with and without SPS according to the Wilcoxon Rank-Sum test. – denotes configurations perform worse with SPS, + denotes configurations that performed better with SPS, and \approx marks no difference between the algorithm with or without SPS.

With respect to HV, nearly all algorithms with SPS monotonically increased with increasing sparsity (Figure A3) during the *effective* runs. The only exceptions were all algorithms under SMOP4, which only decreased marginally between 75% and 95% (Figure A3d). On the flip side, algorithms that did not use SPS tended to generate worse HVs with increasing sparsity, except for a few exceptions. Those exceptions included MOPSO and MOEA/D-DE, which increased modestly with increasing sparsity against SMOP7 and SMOP8 (though the same algorithms with SPS had higher HV in those sparsity ranges) (Figure A3g and Figure A3h). For SMOP[1–5] and SMOP8, all SPS HVs were significantly larger than their non-SPS enabled counter parts at all sparsities (Figure A3). Interestingly, there is a small sparsity window during SMOP6 and SMOP7 where the non-SPS algorithms had slightly better HVs than the SPS enabled algorithms in higher sparsities (i.e., roughly between 56% and 70% sparsity). This small region, along with the relatively lower HV, demonstrates that SPS operates increasingly optimally as sparsities increase.

NDS performance with varying sparsity differed between the various algorithms during the *effective* runs. Most maintained 100% sparsity throughout all sparsity levels. Of the algorithms without SPS, NSGA-II and MOEA/D-DE had non-perfect NDS in SMOP8. Additionally, SPS had slight negative effects on MOPSO and MOEA/D-DE. Increased sparsity boosted variability in MOPSO and MOEA/D-DE for SMOP1, and SMOP4 (Figure A4a and Figure A4d). MOPSO with SPS for SMOP5 and SMOP6 were the only configurations that exhibited a monotonic decrease in NDS with sparsity (Figure A4e, and A4f).



Figure 2. The HV of three types of multi-objective optimization algorithms ran against SMOP1 with and without SPS. SPS consistently increases HV for all algorithms and numbers of decision variables. Note that the variances of some configurations were so small that the confidence intervals were rendered as lines. See Figure A1 for the results of SMOP[2–8].

4.4.2 Generic EA with SPS compared to SparseEA

The most dramatic improvement NSGA-II with SPS had over SparseEA was in run times, as the former ran significantly faster than the later (Table 3 and Table A2; Figure 3 and Figure A6). The only exception to this trend was for SMOP4 with 5000 decision variables, where there was no significant difference in run time between the two algorithms (Table 3, Figure A6d). The log transform of both sets of run times were sub-linear (Figure 3 and Figure A6d), which suggests that both are $o(d^i)|i \in \mathbb{N}$ (i.e., sub-exponential). However, NSGA-II with SPS runs significantly| faster than SparseEA under all 48 scenarios (save SMOP5 with 5000 decision variables) (Table 3 and Table A2). Also, the rate of increase in the SparseEA log-transformed run times increase at a faster rate than the NSGA-II with SPS log-transformed run times (Figure 3 and Figure A6). This suggests that both the asymptotic run time bound of both algorithms are at least polynomial time with respects to *d*, and SparseEA is likely a polynomial of a higher order than NSGA-II with SPS. Furthermore, NSGA-II with SPS runs as much as one order of magnitude faster than SparseEA at 100, and then 3 orders of magnitude at 7500 decision variables (Figure 3).

NSGA-II with SPS demonstrated significantly greater HVs in runs with high numbers of decision variables. NSGA-II with SPS achieved consistently higher HVs throughout the 5000 and 7500 decision variable runs under all test problems (Table 3 and Table A2; Figure 4 and Figure A5), the only exception being SMOP4 at 5000 variables, which did not have significantly divergent HVs. For 2500 decision variables, NSGA-II with SPS and SparseEA on average performed about the same (i.e., three runs with higher HV for SparseEA, three runs with higher HV for NSGA-II with SPS, and two runs with no significant difference in HV) (Table 3 and Table A2; Figure 4 and Figure A5). For the three lower numbers of decision variables, 100, 500, and 1,000, SparseEA achieved significantly higher HVs (Table 3, Table A2; Figure 4 and Figure A5). However, if the HVs are interpolated over the gaps in decision variables, NSGA-II with SPS had a higher HV for the bulk of tested decision variables (approximately 3000 to 7500 decision variables) except for SMOP3 (Figure 4 and Figure A5).

# dec. vars.	Test prob.	Median performance of NSGA-II with SPS			
		HV	Run time	NDS	
100	SMOP1	3.49(3.49)+	2.03(3.89)+	100(100)≈	
100	SMOP2	3.46(3.47) [≈]	1.40(3.70)+	100(100)≈	
100	SMOP3	3.47(3.48)-	$1.51(4.23)^{+}$	100(100) [≈]	
100	SMOP4	3.78(3.78)-	$2.14(4.66)^{+}$	100(100) [≈]	
100	SMOP5	3.77(3.78)-	2.15(4.96)+	100(100) [≈]	
100	SMOP6	3.75(3.78)-	$2.37(4.95)^{+}$	100(100)≈	
100	SMOP7	3.19(3.18)+	$2.16(4.61)^{+}$	$100(100)^{+}$	
100	SMOP8	2.65(2.98)-	1.48(3.76)+	$100(71)^{+}$	
500	SMOP1	3.42(3.44)-	3.25(7.93)+	100(100)≈	
500	SMOP2	3.30(3.36)-	2.99(7.56)+	100(100)≈	
500	SMOP3	3.27(3.42)-	$2.92(7.57)^+$	100(100)≈	
500	SMOP4	3.78(3.78)-	3.49(7.05)+	100(100) [≈]	
500	SMOP5	3.75(3.77)-	3.66(8.68)+	100(100) [≈]	
500	SMOP6	3.72(3.77)	$4.75(9.40)^{+}$	100(100)≈	
500	SMOP7	2.96(3.05)	$3.72(8.74)^+$	100(100)≈	
500	SMOP8	2.31(2.75)-	3.04(7.74)+	$100(100)^{+}$	
1,000	SMOP1	3.40(3.40) [≈]	4.71(13.06)+	100(100) [≈]	
1,000	SMOP2	3.26(3.28)	$4.88(13.01)^{+}$	100(100) [≈]	
1,000	SMOP3	3.15(3.39)-	$4.70(13.35)^{+}$	100(100) [≈]	
1,000	SMOP4	3.78(3.78)-	5.94(9.28)+	100(100)≈	
1,000	SMOP5	3.75(3.75)+	4.86(13.34)+	100(100)≈	
1,000	SMOP6	3.72(3.74)-	6.73(14.81)+	100(100)≈	
1,000	SMOP7	2.88(2.92)	4.96(14.20)+	100(100)≈	
1,000	SMOP8	2.23(2.51)	$4.92(13.12)^{+}$	100(100) [≈]	
2,500	SMOP1	3.38(3.38)+	10.31(48.29)+	100(100)≈	
2,500	SMOP2	3.24(3.24)≈	10.48(48.42)+	100(100)≈	
2,500	SMOP3	3.10(3.38)-	10.26(51.62)+	100(100) [≈]	
2,500	SMOP4	3.78(3.78) [≈]	13.17(15.95)+	100(100) [≈]	
2,500	SMOP5	3.75(3.74)+	10.93(48.72)+	100(100) [≈]	
2,500	SMOP6	3.72(3.73)-	15.31(54.09)+	100(100) [≈]	
2,500	SMOP7	2.83(2.82)+	10.64(52.43)+	100(100) [≈]	
2,500	SMOP8	2.20(2.23)-	10.78(52.75)+	100(100) [≈]	
5,000	SMOP1	3.38(3.12)+	20.13(200.99)+	$100(8)^{+}$	
5,000	SMOP2	3.23(2.95)+	20.57(204.46)+	$100(8)^{+}$	
5,000	SMOP3	3.14(3.06)~	19.76(215.55)+	$100(7)^{+}$	
5,000	SMOP4	3.78(3.68)+	26.25(25.98) [≈]	$100(12)^{+}$	
5,000	SMOP5	3.75(3.60)+	21.85(197.53)+	$100(11)^{+}$	
5,000	SMOP6	3.72(3.63)+	30.55(216.93)+	$100(16)^{+}$	
5,000	SMOP7	$2.82(2.52)^{+}$	21.20(214.22)+	$100(8)^{+}$	
5,000	SMOP8	2.19(1.83)+	20.96(196.42)+	$100(7)^{+}$	
7,500	SMOP1	3.38(3.15)+	29.85(577.32)+	100(9)+	
7,500	SMOP2	$3.23(2.95)^{+}$	30.58(594.09)+	$100(8)^{+}$	
7,500	SMOP3	$3.10(3.09)^{\approx}$	29.56(618.53)+	$100(7)^{+}$	
7,500	SMOP4	3.78(3.68)+	39.50(79.66)+	$100(12)^{+}$	
7,500	SMOP5	3.75(3.60)+	32.42(573.44)+	$100(10)^{+}$	
7,500	SMOP6	3.72(3.63)+	45.69(623.84)+	100(16)+	
7,500	SMOP7	2.81(2.49)+	31.14(608.23)+	$100(8)^{+}$	
7.500	SMOP8	$2.17(1.82)^{+}$	$31.47(657.65)^+$	$100(7)^{+}$	

Table 3: Comparative performance of NSGA-II with SPS and SparseEA.

Let -, \approx , and + denote statistical differences between SparseEA and NSGA-II with SPS according to the Wilcoxon Rank-Sum test. - denotes that SparseEA performed better, + denotes NSGA-II with SPS performed better, and \approx marks no statistical difference between SparseEA and NSGA-II with SPS.



Figure 3. *Comparative* runs which measure the effect of varying number of decision variables on run time for SMOP1. Note that the variances of some configurations were so small that the confidence intervals were rendered as lines. See Figure A6 for the SMOP[2–8] runs.



Figure 4. *Comparative* runs which measure the HV of NSGA-II with SPS and SparseEA over increasing numbers of decision variables for SMOP1. SparseEA suffers significant decreases in HV after 2500 variables. Note that the variances of some configurations were so small that the confidence intervals were rendered as lines. See Figure A5 for the SMOP[2–8] runs.

Finally, NSGA-II with SPS significantly improved the NDS for runs with greater numbers of decision variables (i.e., 5000 and 7,500), with p-values ranging between 3.02e-24 and 1.02e-23 (Table A2; Figure 5 and Figure A7). When the number of decision variables are low (i.e., 100, 500, 1,000, 2,500), NSGA-II with SPS maintains the same number of NDS as SparseEA in most runs, and performs marginally better in some (i.e., SMOP8 with 100 and 500 decision variables). However, when the number of decision variables is increased, SparseEA struggles to maintain a high NDS. Meanwhile, when the number of decision variables is increased to 5000 and 7,000, the runs had median NDS between 16 and 7 (compared to the optimal of 100). These few solutions in the final Pareto set would severely limit the number of Pareto optimal options for a decision-maker. NSGA-II with SPS on the other hand consistently produced 100 NDS in all runs across all numbers of decision variables.

With these performances in mind, there are tradeoffs between the two algorithms. With regards to HV, SparseEA performs better in problems with moderate numbers of decision variables (i.e., between 100 and 1,000). However, as the size of the decision space increases, NSGAII with SPS on average achieves significantly better HV and NDS. In addition, across the all numbers of decision variables, NSGA-II with SPS runs orders-of-magnitude faster than SparseEA.



Figure 5. *Comparative* runs which measure the NDS of NSGA-II with SPS and SparseEA over increasing numbers of decision variables for SMOP1. SparseEA struggles to find solutions with over 10 NDS after 2500 decision variables. Note that many NSGA-II with SPS solutions maintained 100 NDS under all numbers of decision variables, and the median overlaps the perfect NDS line at y = 100. Also, the variances of some configurations were so small that the confidence intervals were rendered as lines. See Figure A7 for the remaining SMOP[2–8] runs.

4.5 Discussion

Considering the simplicity of sparse population sampling, there are remarkable benefits to this elegantly straightforward routine. As the *effective* runs demonstrated, SPS yielded nearuniversal improvements in HV regardless of the test problem. NSGA-II ran particularly well with SPS among the tested algorithms. Not only did it significantly increase HV from NSGA-II without SPS, but it also experienced no losses in NDS, and has the same algorithmic complexity as NSGA-II without SPS (Table 2 and Figure A1). Also, with the exception of a few cases, it consistently found larger HVs than all other algorithms in the *effective* runs (Figure A1). The single exception to this overall trend was NSGA-II without SPS, which had higher HV in low sparsities for SMOP6 (Figure A3f). In the *comparative* runs, NSGA-II with SPS also outperformed the stand-alone LSSMOP SparseEA in high-dimensional problems (i.e., > 5, 000). In this range of decision variables, NSGA-II with SPS outperformed the HV of SparseEA, maintained a consistent NDS, and had a dramatically lower run time. These performance increases highlight the benefits of sparse population sampling. SPS with NSGA-II is simple, stable, and high performing in high-dimensional spaces. However, the relatively diminished performance of NSGA-II with SPS in lower dimensional spaces calls for improvements in further research.

The performance of MOEA/D-DE and MOPSO were less consistent. Most issues were slight NDS losses and instability, which all could plausibly be seen as a worthwhile tradeoff for increased HV. Save for a minority of NDS in the upper twenties, SPS enabled MOPSO and MOEA/DDE were able to keep NDS in the nineties for the vast majority of test runs, which is still a practically useful NDS. This is a plausibly worthwhile tradeoff for the increase in HV. However, there was a fraction of practically unusable cases of NDS losses among SMOP5 and SMOP6 under MOPSO, which warrants further investigation. Also, even though SPS universally improved HV, the SPS enabled algorithms were not able to achieve perfect HV across all scenarios except for NSGA-II with SPS for SMOP4 which achieved near perfect HV (Figure A1, Figure A3, Figure A5). However, it is worth noting that SparseEA was also unable to achieve perfect HV for the entire range of decision variables.

These observed drawbacks are not issues with SPS itself, but with the algorithms that it samples for. We propose SPS not as a complete solution to large-scale sparse optimization problems, but rather as a powerful performance enhancer to incorporate into any future sparse algorithm.

Indeed, it is remarkable that NSGA-II with SPS performs as well as it does, considering no other change was made to the algorithm. Regardless, a more holistic algorithm must be written with SPS in mind in order to address the HV and NDS drawbacks of NSGA-II, MOPSO, and MOEA/D-DE. These improvements will likely be enhanced population sampling, crossover functions, or other critical genetic algorithm operations. Future research should revolve around developing an algorithm that adapts to changes in target sparsity better than the three algorithms in the *effective* runs.

SPS is also fairly sensitive to changes in sparsity. Except for the exceptions discussed in Section 4.4.1 Effects of SPS on standard EAs, sparsity monotonically increased with greater sparsity. Though the high HV at higher sparsity is commendable, future work must develop methods to keep HVs consistently high from low-sparsity problems to high-sparsity problems.

Taking all of these findings into consideration, we provide the following recommendation for solving LSSMOPs. For sparse problems with 100 to 3000 decision variables, SparseEA obtains higher HVs at the expense of longer run times. When dealing with problems that are highly sparse (i.e., roughly 75% sparsity) and large scale multi-objective algorithms (i.e., greater than 3000 decision variables), NSGA-II with SPS is the best option out of the algorithms with respects to HV, run time, and steady NDS.

To summarize, besides the above exceptions, we reject the first global null hypothesis that SPS has no impact or a negative impact on existing algorithms' performance with LSSMOPs (H_0^e). The *effective* runs demonstrated that SPS can be incorporated into and improve upon most existing population-based algorithms with negligible loss to stability and algorithm complexity. There are only one to two parameters (Algorithm 1), which minimizes parametric complexity. It also requires no resources or dependencies from the algorithm beyond the initial population generation, and thus SPS can be combined with any population-based algorithm. We also reject a significant portion our

second main null hypothesis, that SPS with NSGA-II with SPS will perform worse or the same as SparseEA (H_0^c). As observed in the *comparative* runs, as problem complexity increased (between 5000 and 7500 decision variables), NSGA-II with SPS achieved higher HV and NDS. Also, with respect to run time, NSGA-II with SPS performed significantly faster than SparseEA across all configurations. Where SparseEA did outperform NSGA-II with SPS (i.e., the lower decision variable problems) with respect to HV, we are unable to reject H_0^c .

4.6 Conclusions

SPS shows great promise for improving the performance for LSSMOPs. It improves the HV for nearly all algorithms with little to no side-effects and combined with NSGA-II outperforms existing LSSMOP algorithms in high-dimensional decision spaces (as large as 8000 variables). This near-universal improvement entails no known impacts to run time and requires only two parameters. On the other hand, it suffers from minor NDS instability and/or decreases in many test cases, and severe NDS decreases in a small minority of test cases. However, slightly less than optimal NDS is an arguable tradeoff for the remarkable near-universal gains in HV. Future research must focus on how to improve these drawbacks by creating more holistic integrations with population-based algorithms, such as devising customized sparse-variable crossover and mutation operators.

The findings from this study established a solid foundation for the development of a theoretical proof of the proposed sampling strategy. Specifically, future work should address why SPS performs better at higher sparsities, why NDS is critically unstable for MOPSO under SMOP5 and SMOP6, and why NSGA-II with SPS outperforms the other algorithms with SPS. All of these endeavors are critical to making the performance of algorithms with SPS more robust under all sparsity conditions. Nevertheless, SPS brought about near-universal HV improvements with little to

no loss in NDS or run time or change to the main algorithm. Also, this algorithm is one of the few to effectively solve large scale (100 to 8,000-variable) multi-objective optimization problems. By better solving LSSMOPs, SPS shows promise in solving the difficult, large-scale problems throughout engineering

5. AGRICULTURAL INNOVIZATION: AN OPTIMIZATION-DRIVEN SOLUTION FOR SUSTAINABLE AGRICULTURAL INTENSIFICATION IN MICHIGAN

5.1 Introduction

The current global agricultural production system struggles to meet food and nutrition security (El Bilali et al., 2019). This is primarily due to the 30% expected increase in world population by 2050, which will elevate food demand by 50% and require a 70% increase in food production (van Dijk et al., 2021). In addition, food accessibility will be further threatened by adverse climate conditions and volatile food prices, especially in developing and underdeveloped regions of the world (Richardson et al., 2018). Therefore, food production systems, being sensitive to abiotic and biotic shocks, need timely decisions in optimizing resource use. To mitigate these shocks, breeders have developed high yielding resistant crop cultivars. These modern cultivars require optimization of water and nutrient management to minimize yield gaps and to ensure the sustainability of agroecosystems (Tian et al., 2018). Despite being a critical resource in food production, water also faces scarcity due to population growth and climate change. The latest Intergovernmental Panel on Climate Change (IPCC) report highlights increased water-related food scarcity due to the increased frequency of droughts and poor water management (Arias et al., 2021)

Maize (Zea *mays* L.) is the third most important crop after rice and wheat, due to its stable global productivity. This is primarily attributed to high-yielding seeds and modern management practices (Ricciardi et al., 2021). Despite continuous efforts in developing high yielding cultivars, many regions of the world are experiencing a stagnation in yield improvement over the last two decades (Kucharik et al., 2020). One critical component of this stagnation is interannual variation in yield, which is primarily attributed to crop-climate feedback (Coffel et al., 2022; Kukal and Irmak, 2018). To address this variability in yield, a data-driven resource optimization approach is

more feasible with the constrained resources availability (Eeswaran et al., 2021; Kropp et al., 2019). For example, in the United States Corn Belt, farmers were able to boost productivity over the last few decades by optimizing irrigation and nitrogen management (Jin et al., 2019; Wortmann et al., 2017; Zheng et al., 2019).

The traditional method of optimizing agricultural management practices is through field experiments, which are constrained by time and monitory resources (Vilayvong et al., 2015). However, computer modeling, monitoring, and optimization now allow farmers to optimize management practices at greater spatiotemporal scales (Abbasi et al., 2014; Tyagi, 2016). This is critical since managing or sustaining the agroecosystem under short, and long-term climatic variability requires a system approach with complex data handling (da Fonseca et al., 2020). Crop simulation models have been used to investigate the impact of management practices on crop growth and development (Araya et al., 2021; Jha et al., 2018). Additionally, data-driven simulation studies have developed novel methods of resource optimization for management practices at both farm (Kropp et al., 2019) and regional levels (Ban et al., 2019). These advancements are already paying off, considering that precision agriculture and information technology have significantly increased agricultural productivity (Ahmed et al., 2018; Basso, 2020; Monzon et al., 2018). However, system complexity and multi-dimensional data reduce optimization speed (Sabarina and Priya, 2015). In fact, it is impossible to exhaustively examine all management practices to find an optimum. However, optimization techniques efficiently find near-optimal solutions among all possible management practices. Classical optimization techniques, such as gradient descent (Curry, 1944) and linear programming (Chvatal et al., 1983), efficiently solve simple optimization problems, but perform poorly with more complex problems such as multi-modal, non-convex, and nonlinear problems (Deb, 2009). Agricultural optimization falls in the latter category, making it a good candidate for evolutionary optimization (Kropp et al., 2019). This study chose evolutionary optimization because it offers a less specialized but an overall more capable optimization than the classical techniques (Goldberg, 1989). Research has repeatedly demonstrated the efficacy of evolutionary algorithms in developing more optimal crop management practices (Akbari et al., 2018; Groot et al., 2012; Kropp et al., 2019; Mello Jr et al., 2013; Sarker and Ray, 2009).

In addition to evolutionary optimization, the utility of multi-objective (MO) optimization techniques has been widely examined for agriculture management. MO problems contain no single solution that simultaneously satisfies all objectives in the problem (Miettinen, 2012). For example, suppose a farmer aims to minimize irrigation while maximizing their yield. In that case, there are two equally optimal solutions: *Solution A* maximizes yield with considerable irrigation losses, and *Solution B* that minimizes irrigation by not irrigating at all. Each of these solutions is equally optimal, and no better solution exists that outperforms each solution in both categories at once. And between these two extreme solutions typically lie a group of solutions, known as a Pareto front, that are also equally optimal and offer different tradeoffs to the problem (Censor, 1977). In the above example, one solution may use a moderate amount of irrigation while attaining modest yields, which wastes less water than *Solution A* but gets higher yields than *Solution B*. Evolutionary multi-objective (EMO) optimization algorithms specialize in solving multi-objective problems with highly complex and irregular search spaces (Goldberg, 1989), which is why they were the method of choice in the study.

The primary difficulty in optimizing an agricultural system is predicting the weather (Jones et al., 2017). Without proper management, unpredictable meteorological events (e.g., precipitation, minimum air temperature, maximum air temperature, solar radiation) can either help the plant, kill the plant, or produce middling yields. One way to address these uncertainties is to widen the scope

of optimization, since it is better to search innovations that are true for all possible weather patterns than to improve specific management actions for a given year. A data-driven approach to discovering such innovations entails data mining massive datasets of irrigation schedules to determine the defining characteristics of good management practices. Such datasets are hard to obtain through the traditional agronomic literature due to the time and costs required to run agronomic experiments (Vilayvong et al., 2015). However, the combination of crop models and optimization can produce such a dataset (Kropp et al., 2019; Saravi et al., 2021). Given a fixed weather pattern, MO algorithms can generate effective management practices for a given year (Kropp et al., 2019), which can then be data mined for their successful characteristics. This process, known as innovization, uses optimization results to mine data for new innovations in a given system (Deb and Srinivasan, 2006).

This study carefully examines weather patterns and their impacts on an agricultural system over 30 years to prescribe (i.e., innovize) new management actions that achieve sustainable intensification. Sustainable intensification is defined as increasing agricultural yields with no additional impact on environment or land use (Pretty and Bharucha, 2014). This innovization procedure is novel because long-term optimization results are transformed into best practice recommendations without requiring new optimization for future climatic scenarios. Also, innovization has never before been applied to an agricultural application. This study hypothesizes that innovization will generate new knowledge of maize production within a given region. These data-driven innovations will then be translated to easy-to-understand management actions for regional farmers and weather patterns. In order to test these hypotheses, the following objectives were examined against the current best practices that are recommended by the local extension publications (herein called common practices): 1) evaluate the productivity and sustainability of long term innovization-based irrigation scheduling under the regional weather patterns; 2) gauge the productivity and sustainability of long term innovization-based fertilizer applications under regional weather patterns; and 3) gauge the productivity and sustainability of long term innovization-based coordinated irrigation and fertilizer management under regional weather patterns.

5.2 Materials and methods

5.2.1 Modeling overview

Overall, the innovization platform generates optimized management actions for the study site, compares them to management actions following common practices, and finally makes recommended amendments to the common practices accordingly (Error! Reference source not f ound.). We chose a study area with complete meteorological (1989-2018) and soil data to generate the optimized management actions and the management actions following common practices data (Section 5.2.2 Study area). Using a calibrated crop model for the study site, an optimization platform generates the near-optimal management actions for the 30 years of weather data (Section 5.2.3 Experiment setup and 5.2.6 Innovization framework) (Figure A10). Then, an algorithm representing farmers following common practices was established and ran with the same 30 years of weather data (Section 5.2.5 Common practice at the study site). The 30 years of optimized and common practice management actions are then statistically compared (Section 5.2.6 Innovization framework). Specifically, the analysis compares the amount of irrigation applied in each physiological growth stage and the timing of a second nitrogen fertilizer application during the growing season (Section 5.2.6 Innovization framework). Finally, a validation routine measures the performance of the new amended practices and the original common practices to confirm or reject

the efficacy of the innovization procedure. The validation is performed on 14 climatologically similar study sites around the primary study site (Figure A9).



Figure 6: Modeling overview

The study area is a field in the village of Cassopolis within Cass County, Michigan, USA (42° 1' 29.64" N, 86° 10' 15.6" W) (Figure 7). The location is in the humid temperate region of the Midwest USA Corn belt. The long-term (1989-2018) average, minimum, and maximum temperature of the location is 9.81 °C, 3.1 °C, and 16.6 °C, respectively, and the average annual precipitation is 1002.52 mm (PRISM, 2022). About 60 percent of the annual rainfall occurs during the crop growing season of May-October. Although Michigan seems to have abundant freshwater resources, precipitation is low during critical crop growing summer months. Additionally, crops often suffer from water deficits during the peak growing season (Andresen et al., 2011) The major soil type is shallow sandy loam with moderate drainage.



Figure 7. Study site

5.2.3 Experiment setup

5.2.3.1 Crop model

This study employs crop simulation models to create the optimized management practices needed for innovization. The crop model selected for this study was the Decision Support System for Agrotechnology Transfer (DSSAT) version 4.7.5 (Hoogenboom et al., 2017; Jones et al., 2003) DSSAT is a collection of computer modules that are integrated to simulate crop growth for varying soil, weather, and genotype databases. DSSAT is also easily integrated into different modeling packages to consider climate variability and change (Han and Ines, 2017), remote sensing data (Li et al., 2015; Richetti et al., 2019), and advanced machine learning techniques (Singh et al., 2019). Its modules are reconfigurable and reusable, permitting users to modulate the code that connects the state and rate variables (de Abreu Resenes et al., 2019). These sets of capabilities make DSSAT a clear choice for this study.

5.2.3.2 Soil and weather data

Based on the soil type classification of the web soil survey webpage, the dominant soil for the study area was identified as sandy loam (NRCS, 2019). Sub-surface tile drain depth and spacing were kept at 75 cm and 10 m, respectively. Initial soil water content was held at 75% $(0.145 \text{ cm}^3/\text{cm}^3)$, based on soil sampling results from 2018 (Jha, 2019).

Daily weather data (i.e., solar radiation, maximum and minimum temperature, and rainfall) were obtained from Dowagiac, MI, the nearest Enviroweather station. Enviroweather is a weather station network developed and supported by Michigan State University (MSU Enviroweather, 2019). Since the weather station was established in 2014, historical weather data prior to 2014 (1989-2013) were collected from the Global Historical Climatology Network (GHCN) for

Cassopolis to complete the 30-year climatological datasets (Menne et al., 2012). We used these daily weather data to calculate the growing degree days (GDD), a surrogate for crop phenological development. The Maize growing season is May through October.

5.2.4 Prediction of developmental stages based on growing degree days

Maize expresses a determinate growth habit with distinct vegetative and reproductive phases. Maize developmental stages are first based on vegetative development (V) and then on reproductive development (R). The most adopted method for determining vegetative stages is the Leaf Collar method (Ritchie et al., 1986) which can vary with hybrid, initial soil and planting conditions, and location. However, most of the Corn Belt hybrids produce 19-20 leaves. The appearance of consecutive leaves is driven by air temperature and hence can be predicted or tracked by estimating degree days or heat accumulation. The most recommended method of predicting heat accumulation is GDD, popularly known as 30/10 method (Abendroth et al., 2010). The optimal growing temperature for maize starts at 10° C and ends at 30°C. Using 30-year historical weather data, we calculated the degree days (Equation (2) in each year and estimated developmental stages.

$$GDD = \frac{T_{MIN} + T_{MAX}}{2} - 10$$
 (2)

Where, *GDD* is growing degree day in Celsius, T_{MIN} is the minimum daily air temperature, and T_{MAX} is the maximum daily air temperature. Note that if temperatures are lower than 10° C, T_{MIN} is kept constant at 10° C, if temperatures are greater than 30°C, T_{MAX} is kept constant at 30°C.

Temperature based crop vegetative and reproductive development stages were used to optimize irrigation and nitrogen management strategies. Predicted stages based on air temperature can be seen in Table A3. These average values for each predicted stage and day were used in
setting up the constants and constraints for optimization, including when to irrigate (30-year minimum of V6 through 30-year maximum of R2) and when to apply nitrogen (30-year minimum of V6 through 30-year maximum of V14) (Table 4).

All of the innovization years (1989-2018) were divided into dry, normal, and wet years of precipitation. Dry years are the driest 1/3 of the innovization years, wet years were the wettest 1/3 of the innovization years, and the final 1/3 were the normal years (Table A3). These categories were also used to characterize the precipitation for the validation scenarios (Table 4).

5.2.5 Common practice at the study site

Agronomic management of maize is primarily driven by specific developmental stages determined by the heat accumulation in plants (Abendroth et al., 2010). Growers in Cassopolis determine initial crop water requirement by measuring the accumulated soil moisture before the growing season. As the maize develops physiologically, growers irrigate after the midpoint of the vegetative stage (V6-V8) in addition to supplemental rainfall. Physiologically, crops need more water for biomass growth during critical growth stages, and hence this requirement is met through irrigation. Based on these growth stages and in order to optimize water use efficiency, farmers in Cassopolis generally plant on May 15. The irrigation source is groundwater pumping, and the method is sprinkler irrigation. Under normal weather conditions, the common practice is to irrigate 10 mm after 50 days of planting (V6-V8 stage) at 5 days intervals up to the tasseling stage (60-70 days after planting (DAP)). For reproductive stages (i.e., from the tasseling stage onwards), growers irrigate 20 mm at 5 days intervals up to the R2 stage (i.e., blistering stage, or 100-110 DAP) (Figure 8) (Kelley, 2016; Kelley and Hall, 2020; Kranz et al., 2008). In the case of heavy rainfall (i.e., greater than 20 mm), farmers typically wait ten days before resuming irrigation (Figure 8). Similarly, nitrogen application time and amount are important to yield production and potential groundwater contamination. The common practice of applying nitrogen for Southwest Michigan's sandy to sandy loam soils is 200-210 Kg N/ha. This is based on the tool Maximum Return to Nitrogen (MRTN) recommendation, which growers adopt in the Corn Belt of the USA (Gramig et al., 2017). This application is split into two doses: pre-plant incorporation (75%) and another side-dressed dose (25%) between V6 to V14, depending on weather conditions. However, nitrogen requirement increases rapidly after the V8 growth stage (8-leaf collar stage), i.e., 55-60 DAP. The vigorous biomass growth during mid to late vegetative stages requires a large amount of nitrogen. Growers usually confine their side-dress application to V8-V10. Since rain negatively affects the application of nitrogen, the common practice is to delay the application until five days after heavy rainfall events (i.e., greater than 20 mm), three days after moderate rainfall events (i.e., greater than 10 mm), and two days after any other rainfall event (Figure 9). These are the practices that this study aims to improve through the process of innovization.



Figure 8. Flow chart of common irrigation practices in Cass County, Michigan.

5.2.6 Innovization framework

The first task in agricultural innovization is to generate many near-optimal irrigation and nitrogen schedules for a vast number of weather scenarios. These schedules are ideally generated using numerous optimizations runs over the entire span of potential weather conditions; a wide breadth of initial conditions ensures that we are analyzing the widest range of possible meteorological scenarios. The agronomical literature recommends 30 years of weather representative to better understand the cropping systems (Deshcherevskaya et al., 2013). Therefore, from 1989 to 2018, 30 independent optimization runs improved the common management practices (i.e., irrigation and nitrogen applications) for each growing season.

All MO runs optimized three objectives: maximizing maize yield, minimizing water usage, and minimizing nutrient leaching. The decision variables (i.e., the management practices to optimize) included when and how much to irrigate during the growing season, and when to apply the second nitrogen application (Table 4). The first nitrogen application, 150 kg/ha at DAP 0, remains constant throughout all scenarios (Table 4). Formally, the optimization is defined by:

$$\max f_{Y}(i_{46}, \dots, i_{118}, N)$$

$$\min f_{I}(i_{46}, \dots, i_{118}, N)$$
(3)
$$\min f_{L}(i_{46}, \dots, i_{118}, N)$$



Figure 9. Flow chart of common nitrogen practices in Cass County, Michigan

where, i_{46} represents the 30-year minimum start date of V6 (DAP 46), i_{118} represents the 30-year maximum end date of R1 (DAP 118), and *I* is the entire irrigation schedule ($I = \{i_{46}, ..., i_{118}\}$). In other words, each $i_d \in I$ represents the amount of irrigation for day-afterplanting *d* in the irrigation schedule. The total irrigation applied on DAP *i* is between 0 and 30 mm ($i \in [0,30]$). *N*, which represents the date of the second nitrogen application, falls between the minimum V6 and the maximum V14 ($N \in [\min(V6), \max(V14)]$) for the 30 years of study (Table A3) and involves applying 50 kg/ha of nitrogen. f_Y , f_I , and f_L are yield, total irrigation usage, and total nitrogen leached into the soil, respectively. Each of these objectives is a function of the irrigation schedule and nitrogen application variables, as well as the fixed soil, weather, and cultivar for each year. DSSAT provides the total irrigation, yield, and leaching (f_I , f_Y , and f_L) by simulating maize growth for the given soil conditions, cultivar genotype, weather, and management practices at the study site.

The EMO algorithm chosen for this study was the non-dominated sorted genetic algorithm III (NSGA-III), a powerful, many-objective optimization algorithm that simultaneously prioritizes solution convergence and diversity (Deb and Jain, 2014). Because of the stochastic nature of EMO algorithms, the optimization was repeated 30 times for each year with different random seeds. Due to the long run time of the optimization in preliminary runs, thirty repetitions were determined to be the ideal balance between run time and sample size. This lowers the odds that a single optimization with poor convergence is included in the results.

Every irrigation solution ran during the optimization, regardless of fitness, was saved for analysis during the innovization. All solutions for 30 runs for a given year were then combined into a single set and sorted into non-dominated fronts (another term for a Pareto front). Nondominated sorting takes all solutions from all 30 optimization runs and sorts them into Pareto

Variables			
Parameter	Differences between runs		
Weather	Different observed data for each year		
Irrigation	When and how much to irrigate during the growing season between the 30-year minimum of V6 (DAP 46) and maximum start day of R2 (DAP 118)		
Second nitrogen application	25% side dressed between the 30-year minimum of V6 and maximum of V14		
Constants			
Parameter	Constants across runs		
Planting date	May 15th		
First nitrogen application	preplant incorporation 75% (of 200 kg)		
Total nitrogen application	200 kg		
Innovization period	1989 - 2018		
Variables			
Parameter	Differences between runs		
Weather	Different observed data for each year		
Irrigation period Second nitrogen application	Between the 30-year minimum of V6 (DAP 46) and maximum start day of R2 (DAP 118) 25% side dressed between the 30-year minimum		
Constanta	of V6 and maximum of V14		
Constants			
Parameter	Constants across runs		
Planting date	May 15th		
First nitrogen application	preplant incorporation 75% (of 200 kg)		
Total nitrogen application Innovization period	200 kg 1989 - 2018		

Table 4. Optimization variables and constants

fronts with increasing optimality. For example, the first front is globally optimal for the three objectives; the second front is globally optimal if the first front is removed, and so on.

Traditional optimization algorithms tend to struggle to optimize large-scale optimization problems. Luckily, expert knowledge of irrigation helps simplify the optimization procedure: irrigation schedules are known to be sparse (Kropp et al., 2019). In simpler terms, plants physiologically only need irrigation during a minority of days in the growing season. Formally, the optimal *I* is always a vector of values that are mostly zero. This class of problems are known as large-scale sparse multi-objective problems LSSMOP (Zille and Mostaghim, 2017). There are a several options to take advantage of this knowledge. Kropp et al. (2022a) demonstrated that NSGA-II, a version of NSGA-III for lower objective spaces (Deb et al., 2002), solves sparse problems effectively with sparse population sampling (SPS). SPS consists of modifying the initial population of evolutionary algorithms to enforce a target sparsity, which starts the initial population off on advantageous footing during the first generation. The target sparsity for this study was set to 80%, or an irrigation schedule with 80% of days without irrigation based on the sparsity of common practice irrigation schedules in preliminary runs.

Once the 30 repetitions for the 30 years of optimization are completed, we are left with a large dataset of 489,614 unique irrigation schedules, their respective yields, leaching levels, and water usages. However, this dataset still poses a challenge to innovization: no attributes in the dataset are relevant to improving common practices (Figure 8 and Figure 9). Firstly, the raw attributes include the days of the growing season and how much irrigation was applied during those days, which is directly tied to the weather of that given year. Thus, the irrigation schedules must be first transformed into a weather-independent format. In addition, a raw irrigation schedule holds little meaning for a farmer in the study area and transforming the results into relevant

attributes will simplify the results. Therefore, the optimization results were run through a process that transformed the raw irrigation schedules into what we will refer to as a knowledge attribute table. The knowledge attribute table includes the irrigation schedule transformed into variables that describe how much water is applied during each physiological stage. For example, one attribute describes how much irrigation water was applied during the V8 stage of growth. Transforming yearly optimal irrigation schedules into physiological terms standardizes the irrigation schedules across all weather patterns. In addition, it makes the data more understandable to farmers, who can easily determine the current physiological stage of a plant in the field. See the complete list of physiological attributes in Table 5. The final attributes are the threshold of heavy, moderate, and light rain rainfall in the context of nitrogen application. These values correspond to the common practices in the study site to delay the second nitrogen application by five days after heavy rainfall, three days after a moderate rainfall, and one day after light rainfall (Figure 9). To determine the heavy rain threshold from raw irrigation schedules, we select the total rainfall from the day with the greatest amount of rainfall within five days before the second nitrogen application. This represents the amount of rain the optimized nitrogen management was able to tolerate. The same process calculated the moderate and light rain thresholds, but within three and one day(s) before the second nitrogen application, respectively.

With knowledge attributes calculated, we next search for ways to innovate the common practices of Cass County farmers (Figure 8 and Figure 9). Looking for innovations in the common practices of producers makes recommendations easy and understandable for existing practitioners to follow. One area for improvement in the existing system is how much irrigation is applied during each application. As described in Figure 8, farmers in Cass County currently irrigate 10 mm of water during the vegetative phases of maize, and 20 mm during the reproductive stages of maize.

Knowledge attribute code	Description	
w _{V6}	Total water during V6	
W_{V7}	Total water during V7	
W_{V8}	Total water during V8	
W_{V9}	Total water during V9	
W_{V10}	Total water during V10	
W_{V11}	Total water during V11	
<i>W</i> _{<i>V</i>12}	Total water during V12	
<i>W</i> _{<i>V</i>13}	Total water during V13	
W_{V14}	Total water during V14	
W_{R1}	Total water during R1	
m_h	Threshold for heavy rainfall	
m_m	Threshold for moderate rainfall	
m_l	Threshold for light rainfall	

Table 5. Knowledge attributes

These fixed application amounts are excellent targets for innovization since they only consider whether the plant is in the vegetative or reproductive stages. Optimizing irrigation application amount for individual growth stages (i.e., V6 through V14, and R1) would give the farmer better control in managing the water use of their crop. Additionally, an optimized set of application amounts per growth phase would be simple to communicate with farmers. To discover targeted application amounts for each physiological phase, we compared the optimized median values of w_{V6} through w_{R1} from the 30 years of optimized practices to the values of w_{V6} through w_{R1} brought about by 30 years of common practices in Cass County. These 30 years of common practices were realized by first designing an algorithm that simulates the decision-making of a farmer following common practices as described in Figure 8. Let this simulated farmer be known as g^{com} . Then, the simulated farmer runs for all 30 years of weather data, and the of w_{V6} through w_{R1} are calculated from their management decisions. With the computer-optimized and the common-practice-based medians for w_{V6} through w_{R1} , the percentage difference is calculated between the two as follows:

$$\Delta \widetilde{w}_p = \frac{\left(\widetilde{w}_p^{opt} - \widetilde{w}_p^{com}\right)}{\widetilde{w}_p^{com}}$$
(4)

where, $p \in \{V6, ..., V14, R1\}$, \tilde{w}_p is the median water usage during physiological phase p for either common practices (\tilde{w}_p^{com}) or optimized practices (\tilde{w}_p^{opt}), and $\Delta \tilde{w}_p$ is the median difference between common practices and optimized practices ($0 \le \Delta \tilde{w}_p \le 1$). With the set of recommended percentage differences $\Delta W = \{\Delta \tilde{w}_{V6}, ..., \Delta \tilde{w}_{V14}, \Delta \tilde{w}_{R1}\}$, we can then advise the farmer to alter the default irrigation application amount in common practices (i.e., either 10 mm in vegetative stages or 20 mm in reproductive stages) according to ΔW . Formally speaking, we would recommend:

$$i_p^{rec} = i_p + i_p * \Delta \widetilde{w}_p \tag{5}$$

where, i_p is the amount of irrigation applied during the physiological phase p according to common practices (i.e., $i_p \in \{10, 20\}$), and i_p^{rec} is the new recommended irrigation amount for this physiological growth phase. The timing of irrigation would remain intact from common practices. In summary, our recommended irrigation practice would take the form $I^{rec} = \{i_{V_6}^{rec}, ..., i_{V_{14}}^{rec}, i_{R_1}^{rec}\}$, where each element in the vector I^{rec} is the amount of irrigation that is applied during the given physiological phase.

A similar innovization procedure occurs for improving the second nitrogen application. According to the common practices, before irrigating, farmers will wait for five consecutive days after a heavy precipitation event (i.e., greater than 20 mm), three days for a moderate precipitation event (i.e., greater than 10 mm), and two days after any light precipitation events. We hypothesized that these thresholds, denoted as m_h , m_m , and m_l , can be innovated. To test this hypothesis, we examined the precipitation of the days leading up to the optimal fertilizer applications. For each optimal result, we search for what has constituted a heavy, moderate, and light precipitation event in that optimal result, or m_h^{opt} , m_m^{opt} , and m_l^{opt} . In other words, we record the most extreme weather events (i.e., the amount of precipitation on the day with the most rain) five days before the second application, three days before the second application, and two days before the second application. We then take the medians of these three sets to obtain \tilde{m}_h^{opt} , \tilde{m}_m^{opt} , and \tilde{m}_l^{opt} . Then, the percentage difference between m_h , m_m , and m_l in common practices (m_h^{com} , m_m^{com} , and m_l^{com}) and in the optimized results (\tilde{m}_h^{opt} , \tilde{m}_m^{opt} , and \tilde{m}_l^{opt}) will define the deviation between all thresholds as follows:

$$\Delta \widetilde{m}_h = \frac{(\widetilde{m}_h^{opt} - m_h^{com})}{m_h^{com}}$$
(6)

$$\Delta \tilde{m}_m = \frac{\left(\tilde{m}_m^{opt} - m_m^{com}\right)}{m_m^{com}}$$
(7)

$$\Delta \widetilde{m}_{l} = \frac{(\widetilde{m}_{l}^{opt} - m_{l}^{com})}{m_{l}^{com}}$$
(8)

where, $\Delta \tilde{m}_h$, $\Delta \tilde{m}_m$, $\Delta \tilde{m}_l$ are the percentage difference between the optimized rain thresholds and the common practices. From $\Delta \tilde{m}_h$, $\Delta \tilde{m}_m$, $\Delta \tilde{m}_l$, recommendations can be made for an innovative threshold for what constitutes heavy, moderate, and light rain as follows. Together, these three recommendations will be known as the set $M^{rec} = \{m_h^{rec}, m_m^{rec}, m_l^{rec}\}$.

$$m_h^{rec} = m_h^{com} * \Delta \widetilde{m}_h \tag{9}$$

$$m_m^{rec} = m_m^{com} * \Delta \widetilde{m}_m \tag{10}$$

$$m_l^{rec} = m_l^{com} * \Delta \widetilde{m}_l \tag{11}$$

5.2.7 Validation

To validate the innovativeness of I^{rec} and M^{rec} , we measure the performance of four strategies. The first strategy represents farmers following common practices, denoted as the common practice strategy. The other three strategies are innovization based: 1) farmers only following irrigation recommendations I^{rec} (irrigation recommendation strategy), 2) farmers only following nitrogen application recommendations of M^{rec} (nitrogen recommendation strategy), and 3) farmers following both irrigation and nitrogen application recommendations (I^{rec} and M^{rec}) (all recommendation strategy). These strategies will measure the performance of I^{rec} and M^{rec} individually, as well as the recommendations enacted simultaneously. Performance is measured in yield, water efficiency, and nitrogen leaching, which DSSAT simulates, given the weather for the season. These strategies will also demonstrate the sensitivity of the irrigation and nitrogen parameters on our agricultural objectives. We measured these performance metrics under as many validation seasons of weather as possible to ensure statistical rigor. Since there are plentiful weather records in Southwest Michigan, we gathered 420 validation seasons of weather available. This included 30 years of historical weather data (1989-2018), and 14 weather stations in Southwest Michigan (Figure A9) $(30 \times 14 = 420)$. These climatologically similar sites are in Allegan, Bainbridge, Benton Harbor, Berrien Springs, Dowagiac, Fennville, Grand Junction, Hart, Keeler, Lawrence, Lawton, Oshtemo, Scottdale, and South Haven counties. All weather data was

retrieved from MSU Enviroweather and the GHCN. Of these 420 validation seasons, 159 are climatologically dry, 135 are climatologically normal, and 126 are climatologically wet.

With these 420 validations seasons, three statistical tests measure the efficacy of the amended practices: g_I^{rec} versus g_I^{com} , g_M^{rec} versus g_M^{com} , and g_{MI}^{rec} versus g_{MI}^{com} . Under each test, 420 DSSAT simulations of the *common practice strategy* are compared to 420 DSSAT simulations of amended practices according to yield, water efficiency, and leaching. Since the data did not consistently follow a normal distribution, the Wilcoxon Rank Sum test was used (Hollander et al., 2013) to test whether these three-performance metrics were different between the two runs.

5.3 Results and discussions

5.3.1 Performance of irrigation recommendation strategy

Overall, the *irrigation recommendation strategy* (Table 6) increased yields over the *common practice strategy*, with a mean increase of 271.11 kg/ha per year (Table 7). This increase was statistically significant, with a *p-value* of 3.5% (i.e., < 5%) (Table A4). Of the validation seasons, 78.83% of seasons saw gains in yield, and the gains far outweighed the losses. The average gain in yield was 364.64 kg/ha per year compared to the *common practice strategy*, as opposed to an average loss in yield of 93.03 kg/ha (Figure 10). Thus, on a yearly basis, the benefits of the *irrigation recommendation strategy* far outweighed the costs. This is further illustrated when considering how much a farmer is expected to gain over the course of 30 years. To do so, we calculated the average cumulative net increase in yields across all study sites over thirty years (Figure 11). To summarize, a farmer can expect an average increase in yield between 6964.18 kg/ha and 7904.82 kg/ha by the end of 30 years (Figure 11) over the *common practice strategy*. Encouragingly, no farmer experienced an increase below 6288 kg/ha over the *common practice*



Figure 10. The average gains and losses in yield between 1989 and 2018.

strategies during the thirty-year period, and one scenario even experienced a 9720 kg/ha increase in yield. These increases imply that yield is highly sensitive to the irrigation parameters in the innovization framework. In summary, despite a minority of validation seasons with minor decreases in yield, the *irrigation recommendation strategy* consistently results in significantly increased yields compared to the *common practice strategy*. Furthermore, Nandan (2021) supports these assertions by demonstrating the potential impacts of designed irrigation scheduling on corn yields in the Midwest USA under interannual precipitation variability.

Irrigation Recommendations			
Knowledge attribute code	Recommended change from common practices (mm)		
W _{V6}	-9%		
W_{V7}	+31%		
W _{V8}	+13%		
W_{V9}	+50%		
W_{V10}	+60%		
W_{V11}	-0.5%		
<i>W</i> _{<i>V</i>12}	-0.5%		
<i>W</i> _{<i>V</i>13}	+14%		
W_{V14}	+13%		
W _{R1}	-10%		
Nitrogen recommendations			
Knowledge attribute code	Recommended change from common practices (mm)		

-60%

-64%

 $+\infty\%^*$

Table 6. Knowledge attributes recommendations

 m_l

 m_h

 m_m

*A change from 0 to 0.8



Figure 11. Cumulative net increase in yield between irrigation recommendation strategy, nitrogen recommendation strategy, and all recommendation strategy strategies from common practice strategy. The lines' transparent margins represent the 95% confidence interval of that average value

These dramatic improvements in yield came at the expense of water use efficiency. Overall, there was a mean decrease in water use efficiency of 66.45 kg/mm·ha across all scenarios compared to the *common practice strategy*. This decrease was statistically significant, with a *p-value* of 0.3% (i.e., < 5%) (Table A4). Furthermore, 80.67% of scenarios had worse water use efficiency than the *common practice strategy*, and these worse scenarios were of a greater magnitude (on average 86.25 kg/mm·ha worse) than the improved scenarios (on average 4.57 kg/mm·ha improved) (Table 7). These outcomes suggest that water use efficiency is sensitive to the irrigation parameters in the innovization model. Though not the desired result, such a decrease

in water use efficiency in the Midwest is an arguably worthwhile tradeoff for the increased yields brought about by the *irrigation recommendation strategy* (Andresen et al., 2011).

The *irrigation recommendation strategy* improved leaching slightly over the *common* practice strategy, with the mean leaching improving by 0.9 kg/ha per year (Table 7). However, this improvement is not statistically significant since the decrease in leaching had a *p*-value of 64% (i.e., > 5%) (Table A4), suggesting that leaching is only minimally sensitive to the set irrigation parameters in our innovization framework. And even if leaching is not significantly changed, as the *p*-value suggests, this is still a promising outcome since it suggests that the *irrigation* recommendation strategy can increase yields with no impacts on nitrogen leaching. Also, two years (2001 and 2008) had sizeable decreases in leaching, which resulted in an average cumulative net decrease of 26.96 kg/ha in leaching over 30 years (Figure 12). This was most likely caused by the difference in irrigation applications around the second nitrogen application. In the years with significantly decreased leaching, the *irrigation recommendation strategy* consistently applied less irrigation when the irrigation application date was near the second nitrogen application date. This also would explain why yields jumped up in these years as increased nitrogen uptake would help improve yields during this critical physiological phase. However, beyond 2001 and 2008, the cumulative net decrease was relatively unstable, with a maximum reduction in yield of 61.57 kg/ha and a minimum reduction of 8.87 kg/ha. In summary, despite minor improvements, leaching did not significantly change under the *irrigation recommendation strategy*, which is a good outcome considering the significant improvements in yield. This contrasts with the literature, which clearly states that the amount and timing of irrigation greatly influences the nitrate leaching. Altering the irrigation schedule to avoid coincidental N application helps crops in uptaking N during critical crop growth stages and hence lessen leaching (Singh, 2021). Gheysari et al. (2009) have

highlighted the impact of different levels of irrigation on nitrate leaching in corn and emphasized that if irrigation and nitrogen application timing can be managed efficiently on the field, the impact on nitrate leaching will be less without comprising corn yields. Therefore, the irrigation recommendation strategy failed to achieve potential statistically significant improvements in leaching. The climate of the validation season affected the performance of the irrigation recommendation strategy. With regards to yield, there is no statistically significant change from the common practice strategy during dry and normal years, while there is a statistically significant increase in yield during wet years (Table 7, Table A4). This suggests that there may have been a deficiency in the *common practice strategy* during wet years that the *irrigation recommendation* strategy addressed. It also emphasizes the relative sensitivity of yield to the irrigation parameters of the innovization model during wet years compared to dry and normal years. Concerning leaching, climate variabilities brought about little change in the performance of the *irrigation* recommendation strategy: dry, wet, and normal years all had no significant change in leaching from the *common practice strategies* (Table 7, Table A4). Finally, though there was a drop in mean water use efficiency during dry years, the change was not statistically significant (i.e., a *p-value* of 15%) (Table A4). However, water use efficiency significantly decreased from the common practice strategy for normal and wet years. Overall, the innovizations had the least amount of impact during dry years and the most amount of impact during wet years.

Table 7. The overall performance of the various strategies as well as its performance over different climate scenarios. Superscripts +, -, and \approx denote a statistical improvement, statistical

Strategy	Mean yield	Mean leaching	Mean water efficiency (kg/mm·ha)	
	(kg/ha)	(kg/ha)		
	Performance across all years			
Common practice	7054.03	23.24	297.88	
Irrigation recommendation	7325.14+	22.37≈	231.43	
Nitrogen recommendation	7072.43 [≈]	23.28≈	299.81^{pprox}	
All recommendations	7333.21+	22.34 [≈]	231.67	
	Performance during dry years			
Common practice	5934.72	14.08	222.86	
Irrigation recommendation	6080.56^{pprox}	13.33≈	175.81^{pprox}	
Nitrogen recommendation	5937.12 [≈]	13.75≈	224.16^{\approx}	
All recommendations	6104.40^{pprox}	13.23≈	176.43 [≈]	
	Performance during normal years			
Common practice	7716.19	28.22	368.66	
Irrigation recommendation	7902.55^{pprox}	28.53 [≈]	290.51	
Nitrogen recommendation	7735.88≈	28.41≈	371.34 [≈]	
All recommendations	7907.21^{pprox}	28.53 [≈]	290.66	
	Performance during wet years			
Common practice	7757.02	29.46	316.72	
Irrigation recommendation	8256.10+	27.04≈	237.56	
Nitrogen recommendation	7775.88^{pprox}	29.67 [≈]	317.56 [≈]	
All recommendations	8248.21+	27.07≈	237.41^{-}	

worsening, or no statistical change from the common practice strategy, respectively.



Figure 12. The cumulative net change in leaching between irrigation recommendation strategy, nitrogen recommendation strategy, and all recommendation strategy from common practice strategy. The lines' transparent margins represent the 95% confidence intervals of that average

value.

5.3.2 Performance of nitrogen recommendation strategy

Compared to the *irrigation recommendation strategy*, the *nitrogen recommendation strategy* (Table 6) offered no clear benefits or drawbacks. With respect to yield, though there are minor drops in yield between the mean yield of the *nitrogen recommendation strategy* and the *common practice strategy* (Table 7), the two strategies are statistically nearly identical (*p-value* of 90%) (Table A4). The gains of the strategy were smaller in comparison to the losses (40.7 kg/ha in gains versus 57.5 kg/ha in losses) (Figure 10), and only 30% of the scenarios were gains. The above findings demonstrate that yield is insensitive to nitrogen application timing in our innovization framework. Finally, the cumulative 30-year yield across the 14 study sites hovered around 0 kg/ha

change from the *common practice strategies* (Figure 11). In summary, despite consistent losses, there was no statistical change in yield between the *nitrogen recommendation strategy*.

The nitrogen recommendation strategy also had very little change in water use efficiency or leaching from the *common practice strategy*. Nominally, there is a minor mean increase in water use efficiency (1.92 kg/mm·ha) (Table 7), but not enough to be statistically significant (*p-value* of 91%) (Table A4). Leaching similarly had a mean increase between the *common practice* and the *nitrogen recommendation strategies* (23.28 versus 23.24 kg/ha) (Table 7), but not to a statistically significant extent (*p-value* of 96%) (Table A4). Therefore, there is no significant change in environmental impact between the two strategies and leaching and water efficiency were largely insensitive to nitrogen application timing in our framework. Considering this lack of improvements, the *irrigation recommendation strategy* is a clear better option than the *nitrogen recommendation strategy*. Finally, climate has no statistical impact on the performance of the *nitrogen recommendation strategy*. Though minor differences exist in performance across dry, normal, and wet years, none of them are statistically significant. Across the board, the *nitrogen recommendation strategy* has no statistical effect on yield, water use efficiency, or nitrogen leaching.

5.3.3 Performance of all recommendation strategy

The *all recommendation strategy* differs from the *common practice strategy* similarly to the *irrigation only strategy*. There is a statistically significant (*p-value* of 3.2%) (Table A4) mean increase in yield of 279.18 kg/ha per year over the *common practice strategy* (Table 7), and 77% of scenarios resulted in an increase of yields. Of this 77%, the average increase was 349.4 kg/ha, which is slightly lower than the increases from the *irrigation recommendation strategy* at 364.6 kg/ha (Figure 10). On the other hand, the 23% of scenarios resulting in yield losses from the

common practice strategy were on average lower than the *irrigation recommendation strategy* at 67.57 kg/ha (Figure 10). Cumulatively over the 30-year study period, the *all recommendation strategy* modestly outpaces the *irrigation only strategy* by 348.9 kg/ha (Figure 11). However, despite these differences between the *all recommendation strategy* and the *irrigation only strategy*, there no significant difference between the yields of both strategies (*p-value* of 94%). Overall, the performance of the *all recommendation strategy* is statistically identical to the *irrigation recommendation strategy* with respect to yield.

Similar to the *irrigation recommendation strategy*, the *all recommendation* strategy suffered from setbacks in water use efficiency compared to the *common practice strategy*. Specifically, there was a mean decrease in water use efficiency by 66.21 kg/mm·ha (Table 7), which was statistically significant (*p-value* of 0.3%) (Table A4). Compared to the setbacks in the *irrigation recommendation strategy*, the mean water use efficiency in the *all recommendation strategy* was nominally worse (by 0.30 g/mm·ha) (Table 7). However, the two strategies both had identical rates of water use efficiency decrease from the *common practice strategy* (*p-value* of 96%). As described before, this is an arguably worthwhile tradeoff for the increase in yield in the Midwest. With respect to leaching, the lack of improvements or setbacks in the *all recommendation strategy* is identical to the *irrigation recommendation strategy* by all statistical measures (i.e., same means, *p-value* of 97%) (Table A4). The effects of climate on performance are similarly identical to the effects of climate on the *irrigation recommendation strategy*.

Compared to the current best-performing strategy, *irrigation recommendation strategy*, the changes in yield, leaching, and water use efficiency are statistically identical, despite nominal differences between the two. However, if one considers the complexity of the proposed practice and, subsequently, its appeal to farmers, the *irrigation recommendation strategy* is the better

choice. Therefore, though they are statistically identical, the *irrigation recommendation strategy* is arguably the practice worth adopting. Meanwhile, the *nitrogen recommendation strategy* offers no statistical improvement to the *common practice strategy* and thus should not be recommended to farmers.

5.3.4 Study implications

The gains in the *irrigation recommendation* and *all recommendation* strategies are also financially worthwhile improvements over the *common practice strategy*. For the *irrigation recommendation strategy*, a farmer in Cass County would gain on average 6,900 kg/ha in yield over 30 years (with negligible changes in leaching) just by slightly altering their irrigation amounts according to the vegetative stage. To grasp the impact of a 6,900 kg/ha improvement in yield, consider the following example. The average price of corn in the US was ¢12 per kg between 1989 and 2018 (FarmDoc, 2021), and thus a farmer with a 500-hectare farm would gain \$427,612.86 over a thirty-year period. This is a sizable amount increase with a very minimal shift from common practices in the region.

Though the efficacy of this study is bounded by the study area, it can be easily expanded on a regional scale. To do so, Extension Educators would collect the common practices of different regions for different cultivars and soil types. Then, the Extension Educators would search for ways to optimize the common practices for each specific combination of region, cultivar, and soil types. Next, an innovization software platform would analyze how those variables could be improved for the study region. All of these individual rules could then be aggregated into a single decision support tool for a region, where farmers could easily access simple amendments to their common regional practices. Alternatively, in areas with low internet connectivity, Extension Educators could distribute the recommendations through simple pamphlets. Such an optimization platform

excellently melds human decision-making and computer optimization, and outputs results that farmers can actually apply to their fields.

5.4 Conclusions

This study examined the efficacy of agricultural innovization under three different strategies: the *irrigation recommendation strategy*, the *nitrogen recommendation strategy*, and the *all recommendation strategy*. Two of the three strategies (i.e., the *irrigation recommendation strategy* and *all recommendation* strategy) both statistically significantly improved yields with no impact to nitrogen leaching. However, both strategies resulted in loss in water use efficiency. Though the drop was not critically significant for the Midwest, it is an area of improvement in future studies. Future studies should add a water efficiency objective to ensure that the innovization outputs solutions that are near-optimal in water efficiency as well as in yield, total water use, and leaching. Overall, the *irrigation recommendation strategy* is the best option to recommend to farmers, due to its high yield, lack of nitrogen leaching, and relative simplicity.

Another room for growth is finding innovations that are specific to climate variabilities. In this study, the *irrigation recommendation strategy* only improved irrigation during wet seasons (Table 7), with normal and dry seasons exhibiting no statistically significant change from the *common practice strategy*. Though this is a promising outcome, there is room for improvement. Future studies should seek to perform innovizations on specific climate scenarios, which can then be designed into more intelligent amendments to the *common practice strategy*. This is well within reason, since seasonal climate forecasts are skillful enough to predict whether a growing season will be wet, normal, or dry climatologically.

Other areas of future work include ground truth studies and different agricultural conventions. Specifically, though the model was thoroughly calibrated and validated on the study site, future work should attempt to implement the recommended practices in the field in order to validate their efficacy in the real world. Finally, the framework can and should be applied to other agricultural conventions, such as organic or regenerative farming. This is well within the capacity of our proposed framework and would entail changing the decision variables and objectives to those unique to the given agricultural system.

In summary, this study demonstrates the efficacy of agricultural innovization. By considering the common practices of real farmers in a study region and optimizing the practices for 30 years of weather data, we generate practical and effective recommendations for farmers. By following our recommendations, simulated farmers were able to gain a 6,900 kg/ha increase in yield over thirty years of management. Moreover, these gains only incurred negligible changes in leaching levels and increased water use. Such methods are promising within a food system that needs to sustainably increase yields to feed an increasingly large and affluent global population.

6. IMPROVED EVOLUTIONARY OPERATORS FOR SPARSE LARGE-SCALE MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

6.1 Introduction

Optimization algorithms solve problems that were once un-solvable before the advent of modern computers. Some problems are relatively simple to solve by algorithms, such as differential function minimization, or problems with unimodal search spaces. On the other hand, other problems are comparatively challenging, and lack known deterministic algorithms that can solve the given problem in a reasonable amount of time (i.e., NP-Complete problems). To solve these types of difficult problems, researchers developed evolutionary algorithms (EA) (Goldberg, 1989), which employ the concepts of natural selection to find increasingly fit, if not optimal, solutions. Other researchers (Eberhart and Kennedy, 1995) developed particle swarm optimization (PSO) algorithms that effectively mimicked the collective problem-solving ability of groups of individually autonomous animals. Both of these demonstrably successful algorithms are known as population-based algorithms.

Population-based algorithms have the added benefit of effectively solving problems with multiple conflicting objectives, known as multi-objective problems (MOP) (Deb, 2009). The objectives of a MOP cannot be satisfied by a single solution, and instead, algorithms solving MOPs return sets of solutions known as Pareto sets. Each solution of a Pareto set is equally optimal amongst all of the other Pareto solutions within the set and represents a trade-off within the multi-objective problem (Censor, 1977).

However, meta-heuristic algorithms become increasingly incapable with increased problem complexity. One pertinent example is the number of decision variables in a problem, since problem complexity exponentially increases with the increase in decision variables (Tian et al., 2019a). This phenomenon, known as the curse of dimensionality, can affect an algorithm's ability to converge onto an optimal set of solutions, or critically slow down the rate of convergence (Cheng et al., 2017). Formally, the literature agrees that problems with 100 or more decision variables are large-scale multi-objective problems (LSMOP) (Zille and Mostaghim, 2017).

The LSMOP literature, which has flourished over the previous 10 year, takes various approaches to overcome the curse of dimensionality in LSMOPs (Tian et al., 2021b). One increasingly common approach is to take advantage of properties of the resulting Pareto front in the problem, such as solution sparsity (Tian et al., 2019a). In fact, many critical engineering problems contain solutions that are large-scale sparse vectors, such as irrigation optimization (Kropp et al., 2022b, 2019), and portfolio optimization (Branke et al., 2009). Recently, researchers have designed a handful of general-purpose algorithms that seek to solve sparse LSMOPs (Tian et al., 2019a; Y. Tian et al., 2020b, 2020a; Zhang et al., 2021). Despite exciting progress, there is still room for improvement. Specifically, many algorithms still suffer degraded performance when optimizing problems with greater than 500 decision variables (Kropp et al., 2022a; Zhang et al., 2021). Additionally, many of these algorithms are computationally expensive, and/or require large amounts of memory.

A common thread connecting many of the above algorithms (i.e., SparseEA, MP-MMEA, MDR-SAEA, PMMOEA, MOEA/PSL, SparseEA2) is that they all employ a two-layer encoding scheme to represent the sparsity and the real-values of the problem genome. We hypothesized that a small-scale EMO algorithm, equipped with sparse genetic operators, is sufficient to solve sparse LSMOPs effectively and quickly. Specifically, we propose a novel population sampler that boosts the fitness of the initial population, a mutation operator that mutates nonzero genome values as well as genome sparsity, and a crossover operator that appropriately maintains the sparsity of the

parents in the child solutions. All operators can be employed in any small-scale evolutionary algorithm, but we chose to pair it with the non-dominated sorting genetic algorithm II (NSGA-II), due to its demonstrable good performance with sparse genetic operators (Kropp et al., 2022a). We hypothesize that these genetic operators will be sufficient to create an algorithm that performs well with respects to convergence, diversity, speed, and memory.

In the remainder of this paper, Section 6.2 Related Work presents key past studies related to the solution of spare optimization problems. Our proposed evolutionary multi-objective optimization algorithm is described in Section 6.3 Proposed Method: Sparse NSGA-II with details of the customized operators for handling sparse variables. Section 6.4 Results presents the results on test and real-world problems involving cases with as many as 6,400 variables. Finally, conclusions of this extensive study are made in Section 6.5 Conclusion.

6.2 Related Work

Research has followed several broad directions to solve LSMOPs. Over the last few years, the LSMOP scholarship has been divided into three categories: decision variable analysis (e.g., MOEA/DVA (Ma et al., 2015), LMEA (Zhang et al., 2018)), cooperative coevolution (e.g., CCGDE3 (Antonio and Coello Coello, 2013) and DPCCMOEA (Cao et al., 2017)), and problem transformation (e.g., WOF (Zille et al., 2018)) approaches (He et al., 2019; Ma et al., 2021; Tan et al., 2021; Tian et al., 2019b, p.; Zhang et al., 2021). Recently, Tian et al. (Tian et al., 2021b) introduced a new category of novel search space MOEAs, which introduce different operators (e.g., mutation, crossover, population initialization) to the search process (e.g., SparseEA (Tian et al., 2019a), SparseEA2 (Zhang et al., 2021), and NSGA-II with SPS (Kropp et al., 2022a)).

As mentioned before, a subset of LSMOPs contains optimal solutions that are sparse vectors. In other words, a minority of decision variables in the problem are significant to convergence and diversity, and the remaining decision variables are zeroes (Tian et al., 2019a).. In the following paragraphs, we outline the most recent attempts to create a general-purpose sparse LSMOP, as well as some algorithms that attempt to solve further subsets of sparse LSMOPs, such as problems with multimodality and long function evaluations.

The earliest example of a general-purpose sparse evolutionary algorithm was SparseEA, proposed by Tian et al. (2019a). SparseEA introduces new population initialization, as well as novel crossover and mutation operators. SparseEA represents problem genomes as two vectors: one binary and one real. The binary vector encodes which positions of the genome are zero or nonzero, while the real vector represents the nonzero value of the given position if it is a 1 in the binary vector. SparseEA showed considerable improvements over other evolutionary algorithms for small- and large-scale optimization problems, especially for problems ranging between 100 and 1,000 decision variables. However, as the number of decision variables increases, algorithm performance begins to suffer with respect to time, convergence, diversity, and the number of resulting non-dominated solutions in the final generation (Kropp et al., 2022a; Tan et al., 2021; Y. Tian et al., 2020b).

Tian et al. (2019a) also designed the sparse multi-objective optimization problem framework (SMOP) as a benchmark test for sparse LSMOPs. The SMOP framework allows researchers to configure problems automatically with arbitrary numbers of decision variables, objective functions, and target sparsity. The test suite includes eight different tests that each represent different challenges in sparse LSMOPs. Since its creation, SMOP has since become the de facto testing framework for sparse LSMOPs (Kropp et al., 2022a; Tan et al., 2021; Tian et al., 2019a; Y. Tian et al., 2020b; Zhang et al., 2021).

Several improved approaches to solving sparse LSMOPs followed the development of SpareEA. MOEA/PSL uses a framework of two unsupervised neural networks to reduce the search space of the sparse LSMOP: a restricted Boltzmann machine and a denoising autoencoder (Y. Tian et al., 2020b). Using this method, the authors discovered that they could overcome the performance of SparseEA over a large swath of the SMOP test suite (all tests except for all SMOP5 tests, and SMOP4 and SMOP6 with 1,000 decision variables) between 1,000 and 10,000 decision variables. Smaller large-scale SMOP runs were not included in the study (i.e., between 100 and 1,000 variables). Additionally, MOEA/PSL, though slower than other conventional LSMOPs, on average, ran faster than the only existing sparse LSMOP (i.e., SparseEA) with 10,000-decision variable SMOP problems (Y. Tian et al., 2020b).

Taking a different approach, Kropp et al. demonstrated that sparse operators, on their own, can significantly improve the performance of population-based algorithms solving sparse LSMOPs (Kropp et al., 2022a). The proposed operator, which was called sparse population sampling (SPS), modified the default population sampler of a number of algorithms, including genetic, particle swarm, and differential evolution algorithms to guarantee a sparsity range in the initial population (Kropp et al., 2022a). SPS was shown to significantly increase the hypervolume of all algorithms in many configurations of SMOP. NSGA-II with SPS was then shown to outperform SparseEA with SMOP problems with greater than 5,000 decision variables with greater speed. However, the algorithm required target sparsity parameters which significantly affected performance.

Three algorithms, MDR-SAEA, MP-MMEA, and SLMEA address sub problems of sparse LSMOPs. MDR-SAEA, proposed by Tan et al. (Tan et al., 2021), aims to improve the performance of sparse LSMOPs with expensive performance objective function evaluations. This particularly difficult class of problems suffers both from the curse of dimensionality and infeasibly long function evaluations, and thus, requires optimization routines with few objective function evaluations. The first stage of the algorithm, multi-stage dimension reduction (MDR of MDR-SAEA), involves a feature selection by sorting the importance of each decision variable through an initial nondominated sorting routine. Then, a genetic algorithm attempts to optimize the distribution and number of the nonzero values throughout the genome. Finally, the surrogate assistant evolutionary algorithm (SAEA of MDR-SAEA), solves this simplified sparse LSMOP. The results are promising, with MDRSAEA obtaining higher IGD than SparseEA and MOEA/PSL for all test problems between 500 and 1,000 decision variables, except for SMOP5 where SparseEA and MOEA/PSL outperformed MDR-SAEA. MDR-SAEA also requires on average fewer function evaluations than the competing algorithms. The algorithm MP-MMEA focuses on solving multi-modal sparse LSMOPs (Tian et al., 2021a). The main concept behind MP-MMEA is to divide the search population into sub-populations, and to guide each population towards diverse and convergent areas of the search space via guidance vectors. Additionally, population size changes dynamically throughout the search process via a merge-and-divide routine. Another novelty of the publication was a test suite called SMMOP, which represents multi-objective problems that are multi-modal, sparse, and large-scale. MP-MMEA is able to outperform various competing algorithms between 100 and 500 decision variables at varying sparsities. SLMEA (Tian et al., 2022), on the other hand, attempts to improve the performance and convergence of sparse super-large-scale multi-objective optimization problems (i.e., roughly greater than 10,000

variables). Most LSMOPs struggle to run at such a large scale; therefore, SLMEA is designed exclusively with matrix operations to improve its performance on GPUs. Algorithmically, SLMEA is a dimension reduction algorithm that clusters the decision space into multiple groups. The routine then optimizes the clusters, rather than the individual decision variables themselves, thus reducing the number of function evaluations. Through these adaptations, SLMEA is able to run drastically quicker than the competing state-of-the-art algorithms and were able to outperform MOEA/PSL and SparseEA in almost all 1,000,000 decision variable SMOP problems.

Two more recent algorithms take further steps towards solving generic sparse LSMOPs: PM-MOEA, SparseEA2, and S-ECSO. PM-MOEA (Y. Tian et al., 2020a) employs pattern mining to improve sparse LSMOP performance. Instead of attempting to determine the exact position of the nonzero values, PM-MOEA finds maximum and minimum candidate sets of the locations of the nonzero values via pattern mining. Through these sets, the dimensionality of the problem can be dynamically reduced according to the pattern mining outcomes. An additional advantage of these methods is that the algorithms require no additional parameters. Finally, PM-MOEA introduces new crossover and mutation operators for the two-layer genome encoding scheme introduced by SparseEA (Tian et al., 2019a). The combination of these methods leads to an impressive range of improvements over the state of the art, improving performance of all SMOP tests except for SMOP3 for problems including 100 through 10,000 decision variables. Zhang et al. (Zhang et al., 2021) take a slightly different approach and directly improves upon the existing drawbacks of SparseEA. Specifically, the authors observed that many existing sparse LSMOP algorithms focus unequal attention on sparsity maintenance. Therefore, the proposed SparseEA2 algorithm aims to balance the optimization of solution sparsity and the values of the resulting nonzero values. This is accomplished using variable grouping techniques. Between 1,000 and 5,000 decision variables, SparseEA2 is able to improve the performance of several scenarios of SMOP tests over varying scenarios. Finally, a competitive swarm optimization (CSO) algorithm, S-ECSO (Wang et al., 2021), adapts CSO for sparse LSMOPs utilizing two components. The first component is a sparse operator, named the strongly convex sparse operator, which effectively generates sparse solutions throughout the CSO. The second component is an improvement on the normal CSO platform, which seeks to better balance convergence and diversity throughout the search process.

6.3 Proposed Method: Sparse NSGA-II

In this paper, we consider sparse LSMOPs as multi-objective problems having at least 100 decision variables (*D*), of which most are zero at the optimal solution. A solution to such a problem will take the form for **x**, such that $\mathbf{x} = (x_1, x_2, x_D)^T$. The set of positions of x that are optimally zero will be denoted by \mathbf{x}^{z*} , such that $\mathbf{x}^{z*} = \{x^z | x^z \in \mathbf{x}^*, x^z = 0\}$. Conversely, the positions that are optimally nonzero will be \mathbf{x}^{n*} , such that $\mathbf{x}^{n*} = \{x^n | x^n \in \mathbf{x}^*, x^n \neq 0\}$. θ is the sparsity of optimal solutions, such that $\theta = 0$ denotes that an optimal solution is entirely zeros, and $\theta = 1$ denotes that an optimal solution is entirely ones ($\theta = 1 - \frac{\sum \mathbf{x}^* = 0}{D}$). *N* will denote the population size throughout the optimization.

In the following sections, we outline three major updates to the commonly used small-scale evolutionary multi-objective algorithm NSGA-II. The first is an updated population sampler called VSSPS (Section 6.3.1 Varied Striped Sparse Population Sampling), the second is an updated crossover operator called S-SBX (Section 6.3.2 Sparse Simulated Binary Crossover), and the last is an updated mutation operator called S-PM (Section 6.3.3 Sparse Polynomial Mutation). The overall algorithm will be referred to as sparse NSGA-II, or S-NSGA-II for short.

6.3.1 Varied Striped Sparse Population Sampling

The SPS algorithm, as proposed by Kropp et al. (2022a), has three observed areas for improvement:

- 1. Since a uniform distribution determines the position of the nonzero genomes in the initial population, there are no guarantees that any given x^n (i.e., optimal non-zero genome position) in **x** will be set to a non-zero value in any of the initial population members
- When a population is initialized with a random distribution of nonzero values, a given population member may contain nonzero values in the correct position and others in other disparate incorrect positions
- 3. SPS is dependent on two parameters: an upper (b_u) and lower (b_l) sparsity bound on the uniform distribution that determines the position of the nonzero positions.

The first two issues are addressed in our proposed striped sparse population sampling (SSPS) algorithm, and the final issue is addressed in our varied SSPS (VSSPS) algorithm.

To effectively outline SSPS, we first define a sparsity mask (*M*). In the context of this paper, *M* is an $N \times D$ matrix that defines which positions in the initial population will be nonzero or zero values. Positions in *M* that are set to 1 will be set to a nonzero value in the initial population. *M* is then used to render the default population (P^i) (generated by the default population sampler in any population-based optimization algorithm) sparse (P^s) through an element-wise multiplication ($P^i \circ M = P^s$). The SPS algorithm guarantees the sparsity of every population member of M, which we define as the sparsity vector $s = (\theta_1, \dots, \theta_N)^T$, has a sparsity within the target sparsity range (i.e., $b_l \leq \min(s) \leq \max(s) \leq b_u$). This procedure runs only once in the beginning of the EA, and nonzero positions defined in M are not enforced throughout the evolutionary algorithm.

The key idea behind SSPS is to arrange the 1s into cyclical stripes through M, instead of randomly and uniformly distributing the 1s throughout M as per the original SPS algorithm (Figure 13). This design aims to address the first concern with SPS. The stripes also address the second concern with SPS: that one individual in the initial population can have a nonzero value within \mathbf{x}^{n*} , but also nonzero values outside of \mathbf{x}^{n*} . Therefore, the fitness boost of having a nonzero value in \mathbf{x}^{n*} is canceled out by having a nonzero value outside of \mathbf{x}^{n*} . SSPS seeks to avoids this issue, since the 1s in any given individual are localized to one region of the genome, and therefore population members with nonzeros in \mathbf{x}^{n*} have a niche compared to other population members with no nonzero values in \mathbf{x}^{n*} (Figure 14). Furthermore, these population members with no nonzero values in \mathbf{x}^{n*} will theoretically have lower fitness than the average population member (Figure 14). Therefore, SSPS will theoretically have a bubble-up effect where initial population members that have nonzero values in \mathbf{x}^{n*} will have a fitness boost, and those without will have fitness penalty. Therefore, after the first generation, the population will theoretically have solutions that have nonzeros clustered around \mathbf{x}^{n*} .





The arrangement of the stripes throughout M is critical for improving the fitness of the first generation. But before going into detail on stripe arrangement, let us refer to the original randomly and uniformly distributed M from SPS as M^u and the striped M from SSPS as M^s . The sparsity vectors for M^u and M^s will be equivalent, thus ensuring that the target sparsities are still within the predefined lower and upper sparsity bounds. The stripes in M^s are arranged according to the given sparsity vector **s** passed to the main SSPS routine. The sparsity vector **s** is first turned into a width vector, which defines the width (**w**) of the stripe for each row of M^s (**w** = $([D \cdot \theta_1], ..., [D \cdot \theta_N])^T$) (Algorithm 1, line 1). To determine stripe placement, SSPS first divides
the width vectors into cycles (**c**) (Algorithm 1, lines 2—7). Each cycle (c_k) is a contiguous subset of the weight vector, such that the sum of each c_k is less than or equal to *D*. Formally:

$$\mathbf{c} \mid c_k \in \mathbf{c}, c_k \subseteq \mathbf{w}, \sum c_k \leq D$$

Finally, for each row i (i.e., individual in the population) of M^s , a stripe of 1s with a width w_i is placed in M^s , such that it is to the right of the stripe in the $(i - 1)^{th}$ row (Algorithm 1, lines 8—28, Figure 13 and Figure 16). If at the beginning of a new cycle, the stripe will go at the 1st column of M^s .

			x_1^n		x_2^n									x_3^n	x_4^n	
Individuals Likely to	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
have HIGH fitness	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Likely to have LOW	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
fitness	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:

Figure 14: A simplified example of how SSPS causes solutions with stripes over optimal nonzero positions (highlighted green) to have greater fitness than those with no overlap with optimal nonzero positions (highlighted red).

```
Input: N, D, P^i, s
   // N \rightarrow population size
   // D \rightarrow # of decision variables
   // P^i 
ightarrow default sampled population
   // {\bf s} \rightarrow Target sparsity vector
   Output: P^s
   // P^s\,\rightarrow\,{\rm the} initial population
   /* ----- FIGURE OUT CYCLES -----
                                                          */
   // Determine the width of each stripe
 1 \mathbf{w} = (\lfloor D \cdot s_1 \rfloor, \ldots, \lfloor D \cdot s_N \rfloor)
 \mathbf{2} \mathbf{c} \leftarrow []
   // Determine # individuals per cycle
 3 l \leftarrow 1 while l \leq N do
      // Find # of sequential widths
           fitting in this cycle
       f \leftarrow packSequentialStripes(\mathbf{w}, l, D)
 4
       // Append new cycle
      \mathbf{c}.append(w[l, l+f])
 5
       // Go to the start of next cycle
 6
     l \leftarrow l + f + 1
 7 end
   /* ----- DRAW STRIPES -----
                                                          */
   // For each cycle
 s k \leftarrow 1 // Cycle counter
 9 i \leftarrow 1 // Individual counter
10 M^s \leftarrow 0_{N,D} // Sparsity matrix
11 for k \in [1, |\mathbf{c}|] do
12
      c_k \leftarrow \mathbf{c}[k]
       // Do stripes fill full cycle?
      gapSize \leftarrow \lceil \frac{\tilde{D} - \sum c_k}{|c_k|} \rceil
13
       // For each stripe in the cycle
14
       position \leftarrow 1
15
       while position \leq D do
16
           if k \neq |\mathbf{c}| then
17
               endPosition \leftarrow position + c_k + gapSize
           else
18
           | endPosition \leftarrow position + c_k
19
           end
20
           if endPosition > D then
21
              endPosition \leftarrow D
22
           end
23
           // Draw stripe for individual
           M^{s}[i, position : endPosition] \leftarrow 1
24
           position \gets endPosition + 1
25
26
           i + +
27
      end
28 end
29 P^s \leftarrow P^i \circ M^s
30 return P^s
```

Algorithm 2: SSPS

Frequently, the cumulative width of stripes does not equal D (Figure 15). This poses an issue, since leaving this difference, or gap, at the beginning or end of the genome will create a

significant bias against those genome positions. Therefore, we need some way of rectifying those gaps throughout the stripes of each given cycle. Generally, our first goal in rectifying gaps should be to get as close to a uniform distribution of nonzero values throughout the population as possible. This will ensure that no nonzero position will be biased positively or negatively in the initial population. The second goal should be to produce individuals with sparsities as close to s as possible. In this study, we lengthen the width of enough stripes to fill in each gap (Algorithm 1, lines 13 and 17). This works well since it first covers all genome positions evenly and without a gap throughout each cycle (Figure A11 in the appendix). And secondly, since as many stripes are packed into each cycle as possible, each gap is relatively small compared to the stripe sizes, and the target sparsity is negligibly affected (Figure A12). The only exceptions to these rules are for sparsities that are very large (i.e., between 0.25 and 0.5) and when in the last cycle in *M*. The latter issue arises because the final cycle often does not have enough remaining stripes from w to fill up *D*, and the resulting sparsities are negatively affected by significant widening. Therefore, in the last cycle, gaps are filled with empty space between each stripe (Algorithm 1, line 19).

SSPS can be configured in two ways: with a single target sparsity (**s** of identical sparsities), or with a range of sparsities. For the latter scenario, **s** is set as a linear interpolation of $|\mathbf{s}|$ points between b_l and b_u . However, to reduce the dependence on parameters, we devised a parameterless version of SSPS called varied SSPS (VSSPS). In VSSPS, b_l and b_u are set to a range of 0.25 to 0, which covers all of the sparsities in the existing sparse LSMOP literature enumerated above. SSPS runs with a run time of O(ND), and takes O(ND) in memory (Algorithm 1).

	I	Rectifi	ed stri	pe gap	2													
S	_				\backslash				\boldsymbol{M}									
$\theta \approx 3/16$		1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
$\theta = 3/16$		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	
$\theta = 3/16$		0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	Cycle of length 5
$\theta = 3/16$		0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	
$\theta = 3/16$		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
$\theta = 3/16$		1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Þ	
$\theta = 3/16$		0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Stripe of width 3
$\theta = 1/8$		0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	widdii 5
$\theta = 1/8$		0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	Stripe of
$\theta = 1/8$		0	0	0	0	0	0	0	0	0	0	1	1	€0	0	0	0	width 2
:		:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	

Figure 15: A simplified example of a striping pattern for a $16 \times N$ mask *M* following the sparsity defined in **s**. Though this genome is not large-scale for illustrative purposes, the same concept applies to any $M \times D$ mask.

6.3.2 Sparse Simulated Binary Crossover

On its own, SBX tends to drive population sparsities towards dense solutions (Figure 16). This is likely because SBX treats zero and nonzero genome positions identically (Deb and Agrawal, 1995). For example, suppose one parent has a nonzero value at position d in the genome and the other parent had a zero value at position d. SBX will create child solutions that are adjacent to the parent values, proportional to the proximity of the two parents' real values. In other words, SBX will typically make both of these positions nonzero values, subsequently making the genome denser. Our algorithm, Sparse SBX (S-SBX), aims to keep the best parts of SBX, while improving its performance with sparse problems. In other words, since sparsity is a critical phenotype of

optimal solutions in sparse LSMOPs, the sparsity of children in S-SBX should be approximately the same as the parents.



Figure 16: The effects of SBX on NSGA-II solving a sparse MOP. There is a strong tendency for the algorithm to create increasingly dense populations.

The flow of S-SBX is as follows. First, all positions of the genome that are nonzero in both parents or are zero in both parents (which we will denote as x^m), are crossed over with normal SBX (Figure 17 Phase 1, and Algorithm 2, lines 1—4). This ensures genome positions containing nonzero values in x^{n*} will get further optimized towards their optimal nonzero real value. In the second phase, S-SBX handles mismatched positions of the genome (i.e., one parent is nonzero and the other parent is zero) (Figure 17 Phase 2, and Algorithm 2 lines 5—8). In this scenario, this subset of the genome (which we will denote as x^m) will be swapped, genome position by genome

position, at random between the parents (Figure 17). Which positions of x^m are swapped is determined by a swap mask vector **v**, defined as:

$$\mathbf{v} \mid \mathbf{v} \in \{0,1\}^{|x^{m}|}$$

Each element of \mathbf{v} is selected randomly from a uniform random distribution. Zeros in \mathbf{v} indicate that the value from parent 1 should go to child 1, and the value from parent 2 should go to child 2 (Figure 17 Phase 2). On the other hand, ones in \mathbf{v} indicate that the values should be swapped, such that the value from parent 1 goes to child 2, and the value from parent 2 goes to child 1 (Figure 17 Phase 2). This phase of the crossover procedure passes on genetic information about the positioning of non-zero values.



Figure 17: A simplified example of sparse SBX. In the first phase, genome positions that are both zero or nonzero are crossed over with SBX. In the second phase, the genome positions that are mismatched are swapped at random in the children genomes. The aim to crossover the best

aspects of the parent genotypes while approximately maintaining parent sparsity.

Input: P_1 , P_2 , p_c , η_c // P_1 ightarrow parents to be mated with P_2 // $P_2 \rightarrow$ parents to be mated with P_1 // $p_c \rightarrow$ probability of crossover // $\eta_c \rightarrow$ distribution of crossover PDF **Output:** C_1 , C_2 // $C_1 \rightarrow$ 1st children from P_1 & P_2 // C_2 ightarrow 2nd children from P_1 & P_2 /* -SBX ON 0/0, and $\not 0/ \not 0$ POSITIONS-*/ 1 $Mask_{n1} \leftarrow P_1 \neq 0; Mask_{z1} \leftarrow P_1 = 0$ 2 $Mask_{n2} \leftarrow P_2 \neq 0; Mask_{z2} \leftarrow P_2 = 0$ // Positions that are 0 & 0 or \emptyset & \emptyset 3 $Mask_m = (Mask_{n1} \land Mask_{n2}) \lor (Mask_{z1} \land Mask_{z2})$ // Run SBX on these positions 4 $C_1[Mask_m], C_2[Mask_m]$ $\leftarrow \mathbf{SBX}(P_1[Mask_m], P_2[Mask_m], p_c, \eta_c)$ /* -RANDOM SWAP 0 & Ø OR Ø & 0-*/ 5 $Mask_{pl} \leftarrow \neg Mask_m$ // Empty swap vector 6 $\mathbf{v} \leftarrow \mathbf{0}_{1,|Mask_{yt}|}$ // Uniformly distribute 1s such that // each position has a 50% of being 1 7 $\mathbf{v} \leftarrow addOnes(\mathbf{v}, \frac{|\mathbf{v}|}{2})$ // Swap genes according to ${f v}$ 8 $C_1[Mask_{pri}], C_2[Mask_{pri}]$ $\leftarrow swapGenes(P_1[Mask_{pr}]), P_2[Mask_{pr}], \mathbf{v})$ 9 return C_1 , C_2

Algorithm 3: S-SBX

As mentioned earlier, the goal of S-SBX is to effectively pass on key genetic traits between the two parents while maintaining the approximate sparsity of the parents. To do so, we create two randomly generated sparse parents with the same sparsities, and then repeatedly perform S-SBX on the original parents while recording the resulting sparsities. Ideally, the algorithm should maintain a tight distribution around the original sparsities of the parents. This has been confirmed through repeated trial runs (Figure A13). The algorithm has a run time complexity of O(ND), and takes up O(ND) in memory.

6.3.3 Sparse Polynomial Mutation

A mutation operator should ideally be the driving force of evolution. Therefore, in the case of sparse LSMOPs, the mutation should drive the population towards the optimal target sparsity of the problem at hand. However, the most common mutation operator in real EA (i.e., polynomial mutation (PM)) falls short of achieving this goal. For context, in polynomial mutation, each genome position has a predefined probability of mutation. If a position is mutated, its altered by a polynomial probability density function (Deb, 2009). This implies that sparsity may be altered if a genome position set at zero is mutated to a nonzero value, or vice-versa. However, these changes in sparsity are theoretically rare. If, say, the probability of mutation is set to 0.1, then there's only a small chance that a position will be flipped from a nonzero to zero or vice-versa. Furthermore, it's exceedingly unlikely that a nonzero value will be flipped to a zero value (Kropp et al., 2022a). Therefore, we can deduce that the ability of PM to mutate sparsity is very subtle, and slightly biased towards denser solutions.

We propose a sparse polynomial mutation operator (S-PM) to address these shortcomings. The goal is to maintain the good performance of the original PM on nonzero values while also effectively mutating solution sparsity. In the first phase of S-PM, PM is run on all of the nonzero positions in the genome, which maintains the proven efficacy of PM on real valued MOOs (Algorithm 3, lines 1—3). In the second phase, the global sparsity of the whole genome runs the possibility of mutating, depending on a parameter p_m^s (Algorithm 3 lines 4—13). If the genome is chosen for mutation, then the sparsity of the genome (θ) is mutated to the value θ' using the same polynomial distribution function from PM. In the case that $\theta' < \theta$, random nonzero positions are flipped in the genome to zeros (Algorithm 3, lines 7—9). In the alternative case where $\theta' > \theta$, then the genome will become denser by flipping the required number of zero positions to plausible nonzero values (Algorithm 3, lines 10—12). This routine aims to create more evolutionary randomness with respect to sparsity while maintaining the mutation of the nonzero portions of the genome. It takes up O(ND) time, and O(ND) memory.

Input: P, p_m, η_m // $P \rightarrow$ input population // p_m \rightarrow probability of mutation // η_m ightarrow distribution of mutation PDF **Output:** P_m // $P_m \rightarrow$ mutated population $1 P_m \leftarrow P$ /* -- MUTATE NONZEROS WITH PM --*/ 2 $Mask_n \leftarrow P \neq 0$ $P_m[Mask_m] \leftarrow PM(P[Mask_n], p_m, \eta_n)$ /* ----- MUTATE SPARSITY -----*/ 4 s \leftarrow getSparsities (P_m) 5 $\mathbf{s}_m \leftarrow \mathbf{PM}(\mathbf{s}, p_m, \eta_n)$ 6 for i^{th} member of P_m do if $s_{mi} < s_i$ then 7 // Flip enough nonzeros to zeros to match s_{mi} $P_{mi} \leftarrow \text{zeros2nonzeros}(P_{mi}, s_{mi})$ 8 end 9 10 if $s_{mi} > s_i$ then // Flip enough zeros to nonzeros to match s_{mi} 11 $P_{mi} \leftarrow \text{nonzeros}2\text{zeros}(P_{mi}, s_{mi})$ end 12 13 end 14 return P_m

Algorithm 4: S-PM

6.3.4 Experimental setup

As mentioned in the introduction, we seek to determine whether modifying the genetic operators of a small-scale, multi-objective optimization algorithm (i.e., NSGA-II in this study) is sufficient for solving sparse LSMOPs. Formally, we define our null hypothesis as H0, which posits the proposed algorithm (S-NSGA-II) has no notable improvement in solving sparse LSMOP over the current state-of-the-art sparse LSMOP algorithms. One exception is for the number of non-dominated solution metric, where our null hypothesis states that S-NSGAII performs strictly worse than the other state of the art algorithms, since it is common to have a number of nondominated solutions that is equal to the population size. In order to accept or reject H_0 , we run both state-of-

the-art algorithms and S-NSGA-II against a battery of tests to test for improvements in performance. Since none of the results followed a single distribution, the Wilcoxon Rank-Sum test was used for significance testing (Hollander et al., 2013) with a significance threshold of 2.5%. This battery of tests includes the eight SMOP benchmark tests (Tian et al., 2019a), as well as two real-world application problems, including neural network (NN) training and portfolio optimization (PO). The goal of the NN training optimization is to find the best weights of a hidden layer of neurons with respects to model complexity and training error (Tian et al., 2019a). The code is included in PlatEMO 3.4, and includes two parameters: size of the hidden layer, and which dataset to choose from the UCI machine learning repository (Dua and Graff, 2017; Tian et al., 2017). In the PO problem, an investor is trying to minimize risk and maximize profits for a large investment portfolio (Y. Tian et al., 2020b). This test problem is also from PlatEMO 3.4 (Tian et al., 2017). As for which algorithms to benchmark S-NSGA-II against, since S-NSGA-II is a generic sparse LSMOP algorithm, we chose four generic state-of-the-art algorithms for sparse LSMOPs: SparseEA2, MOEA/PSL, and PM-MOEA.

Each of the SMOP benchmark problems were ran with 100, 200, 400, 800, 1,600, 3,200, and 6,400 decision variables. We chose to double the number of decision variables with each subsequent configuration in order to observe the asymptotic behavior of the given algorithms as the number of decision variables (i.e., problem complexity) increases. The target sparsity for SMOP was set at $\theta = 0.1$. After preliminary time tests, each configuration was repeated 30 times in order to balance a manageable simulation run time and statistical rigor. Algorithms that failed to solve 30 repetitions for a given configuration within 24 hours was terminated and excluded from results. Three separate benchmarks were included for each configuration: hyper-volume (HV), number of non-dominated solutions (NDS), and run time in seconds. HV is useful in that it covers both the diversity and the convergence of a Pareto optimal front. NDS describes the number of non-dominated solutions in a final population (i.e., the size of the Pareto front) (Kropp et al., 2022a). Ideally, it should be as close to the population size (N) as possible, since having a larger Pareto front (i.e., higher NDS) is more useful to a decision maker than a smaller Pareto front. This is a useful metric for sparse LSMOPs, since many sparse LSMOPs struggle to maintain sufficiently high NDS as the number of decision variables increases (Kropp et al., 2022a). Finally, run time measurements are critical to determining whether or not an algorithm can run within a feasible amount of time, especially as the number of decision variables increases. All optimization runs were executed using the PlatEMO 3.4 (Tian et al., 2017) framework under MATLAB 2022a (MATLAB, 2022). Each competing algorithm took its default parameter from PlatEMO 3.4. The tests were run on 2, 12-core Xeon processors with 125GB of RAM.

Each SMOP hypothesis is denoted by the following nomenclature: H_0^T :*U*/SMOP*L*, such that *U* is the number of decision variables, *T* is the test metric used, and *L* is the SMOP test for the given hypothesis. The metrics include *HV*, *NDS*, and *RT* (run time). For the application tests, we denote the NN and PO-based hypotheses as H_0^T :*U*/NN_D*K* H_0^T :PO_D*K* respectively, where *E* is the size of the hidden layer (for NN only), and *K* is the data set used. Let * denote a wildcard that matches any value for *U*, *L*, *K*, or *E*.

6.4 Results

6.4.1 SMOP Results

On the whole, S-NSGA-II ran faster than the existing methods, resulted in better HV, and maintained consistent NDS throughout the different scenarios. First off, S-NSGA-II achieved universally better HV over the majority of scenarios (Table 8), and the exceptions are as follows. S-NSGA-II struggled to outperform the base algorithms for SMOP4 problems. SparseEA, SparseEA2, and PM-MOEA all outperformed S-NSGA-II with test problems with 100 and 200 decision variables (Figure A17). Furthermore, PM-MOEA outperformed S-NSGA-II with SMOP4 under any number of decision variables. On the other hand, S-NSGA-II eventually outperforms SparseEA, SparseEA2, and MOEA/PSL in SMOP4 in higher decision variables (Figure 18). This relatively poor performance of S-NSGA-II at lower decision variables is a trend throughout the other test problems. Also, the relatively worse performance of S-NSGA-II at solving SMOP4 demonstrates an area for improvement in S-NSGA-II (Figure A17). The only distinct property of SMOP4 is low intrinsic dimensionality (Tian et al., 2019a), suggesting that S-NSGA-II needs to better address this class of problem at lower dimensions. SparseEA2 is also able to outperform S-NSGA-II for the majority of SMOP problems between 100 and 200 decision variables (Figure 18, Figure A14—Figure A21). S-NSGA-II is also able to achieve the best HV for SMOP5 and SMOP6 (Figure A18, Figure A19), until MOEA/PSL eventually outperforms it in higher dimensional spaces, suggesting that MOEA/PSL excels at solving unimodal sparse large-scale problems (Tian et al., 2019a).



Figure 18: Typical mean HV (i.e., performance) of S-NSGA-II compared to state of the art generic spares LSMOPs. In this scenario, the algorithms are solving SMOP3 at different number of decision variables. Note how the performance of all of the algorithms slowly decline with increasing numbers of decision variables. SparseEA and SparseEA often perform very well in the relatively low decision spaces, and then experience serious drops in HV around 3,000 decision variables. S-NSGA-II often outperforms the majority of algorithms in large decision spaces. The shaded areas around each mean line are the 95% confidence interval for the mean.

Table 8: Performance of different algorithms compared to sparse NSGA-II in solving smop problems. Performance is measured in hypervolume, run time, and number of non-dominated solutions. The performance of S-NSGA-II is the first number in each cell, followed by the base algorithm in parentheses. Scenarios where S-NSGA-II performed better, worse, or approximately the same as the base method are denoted with +, −, and ≈ respectively. Scenarios where the base method did not halt within a reasonable amount of time are denoted with —.

prob HV Time NDS HV Time NDS HV Time NDS 100 SMCD 558(355 005(100) 058(055) 251(349) 0000(1000) 058(055) 251(349) 0000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 071(12) 1000(1000) 058(057) 101(15) 071(12) 1000(1000) 058(057) 101(15) 071(12) 1000(1000) 058(057) 101(15) 071(12) 1000(1000) 058(057) 101(15) 071(12) 1000(1000) 058(057) 101(15) 071(12) 110(12) 1000(1000) 058(057) 101(15) 071(12) 110(12) 1000(1000)	D	Test	S-NSGA-II vs SparseEA		S-N	SGA-II vs Sparse	FA2	S-N	SGA-II vs MOE	A/PSL	S-NSGA-II vs PM-MOEA			
$ \begin{array}{ $	D	prob	HV	Time	NDS	HV	Time	NDS	HV	Time	NDS	HV	Time	NDS
100 SMOP2 0.580.57 0.97(1.4) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 0.97(1.5.2) 100.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000) 0.580.57 10.00(1000)	100	SMOP1	0.58(0.58)*	2.53(4.08)*	$100.00(100.00)^{=}$	0.58(0.58)	2.53(3.91)*	$100.00(100.00)^{=}$	0.58(0.58)*	2.53(4.96)*	$100.00(100.00)^{=}$	0.58(0.58)*	2.53(12.66)*	$100.00(100.00)^{=}$
100 SAMC3 0.584C37 0.974-75 1000X(1000) 0.584C37 0.944C37 1000X(1000) 0.544C37 0.944C37 1000X(1000) 0.944C37 1000X(10000) 0.944C37	100	SMOP2	0.58(0.56)*	0.97(4.10)*	100.00(100.00)=	0.58(0.58)	0.97(3.89)*	100.00(100.00)=	0.58(0.57)*	0.97(5.32)*	100.00(100.00)=	0.58(0.57)*	0.97(13.42)*	100.00(100.00)=
100 SAMP4 822023 0.000(1000) 622023 0.000(1000) 622023 0.000(1000) 622023 0.000(1000) 622023 0.000(1000) 622023 0.000(1000) 622023 1.000(1000) 622023 1.21531 1.000(1000) 622023 1.21531 1.000(1000) 622023 1.21531 1.000(1000) 622023 1.21531 1.000(1000) 622023 1.21531 1.000(1000) 623023 1.21531 1.000(1000) 623033 1.21531 1.000(1000) 623033 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 1.000(1000) 623035 1.21531 <th< td=""><td>100</td><td>SMOP3</td><td>0.58(0.57)*</td><td>0.97(4.75)*</td><td>100.00(100.00)=</td><td>0.58(0.58)=</td><td>0.97(4.54)*</td><td>100.00(100.00)=</td><td>0.58(0.58)=</td><td>0.97(4.78)*</td><td>100.00(100.00)=</td><td>0.58()+</td><td>0.97(—)*</td><td>100.00()*</td></th<>	100	SMOP3	0.58(0.57)*	0.97(4.75)*	100.00(100.00)=	0.58(0.58)=	0.97(4.54)*	100.00(100.00)=	0.58(0.58)=	0.97(4.78)*	100.00(100.00)=	0.58()+	0.97(—)*	100.00()*
100 SMOP5 6.820.82) 2.00.821.020 6.820.82) 2.00.821.000 6.820.82) 2.851.871 0.00.810.000 6.820.82) 2.851.871 0.00.810.000 6.820.82) 2.851.871 0.00.810.000 6.820.82) 2.851.871 0.00.810.000 6.820.82) 2.851.871 0.00.810.000 0.550.873 0.00.810.000 0.550.873 0.00.810.000 0.550.875 0.151.971 0.00.810.000 0.550.875 0.151.971 0.00.810.000 0.550.875 1.121.8301 0.00.810.000 0.550.875 1.121.8301 0.00.810.000 0.550.875 1.121.81.901 0.00.810.000 0.550.855 1.182.1351 0.00.810.000 0.550.855 1.182.1351 0.00.810.000 0.550.855 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0.00.810.000 0.550.851 1.182.1351 0	100	SMOP4	0.82(0.82)	0.94(4.37)*	100.00(100.00)=	0.82(0.82)	0.94(5.23)*	100.00(100.00)=	0.82(0.82)	0.94(6.03)*	$100.00(100.00)^{=}$	0.82(0.82)	0.94(20.83)*	100.00(100.00)=
100 SMOP6 0.83(0.82) 1.25(4.98) 100.00(10.00) 0.83(0.82) 1.25(5.81) 10.00(10.00) 000 SMOP6 0.84(0.32) 0.25(4.98) 0.000(10.00) 0.34(0.33) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.04(2.32) 0.000(10.00) 0.34(0.33) 0.04(2.32) 0.04(2.32) 0.000(10.00) 0.34(0.33) 0.04(2.32) 0.000(10.00) 0.34(0.33) 0.04(2.00) 0.34(0.33) 0.04(2.00) 0.34(0.33) 0.04(2.00) 0.000(10.00) 0.56(0.54) 1.12(3.66) 10.000(10.00) 0.56(0.54) 1.12(3.56) 10.000(10.00) 0.55(0.52) 1.12(3.56) 10.000(10.00) 0.52(0.52) 1.23(5.82) 10.000(10.00) 0.52(0.52) 1.23(5.82) 10.000(10.00) 0.52(0.52) 1.24(5.84) 10.000(10.00) 0.52(0.52) 1.12(3.56) 10.000(10.00) 0.52(0.52) 1.12(3.56) 10.000(10.00) 0.52(0.52) 1.24(1.58) 10.000(10.00) 0.52(0.52) 1.24(1.58) 10.000(10.00) 0.52(0.52) 1.24(1.58) 10.000(10.00) 0.52(0.52)	100	SMOP5	0.82(0.82)*	0.96(4.88)*	100.00(100.00)=	0.82(0.82)*	0.96(5.28)*	100.00(100.00)=	0.82(0.81)+	0.96(7.19)*	100.00(100.00)=	0.82(0.82)+	0.96(15.69)*	100.00(100.00)=
100 SMOP? 0.34(3.3) 0.94(4.40) 100.00(100.00) 0.34(3.57) 0.094(1.57) 10.00(100.00) 0.49(3.57) 0.00(100.00) 0.49(3.57) 0.00(100.00) 0.49(3.57) 0.12(1.53) 100.00(100.00) 0.94(3.57) 1.12(1.51) 100.00(100.00) 0.99(-7) 100.00(100.00) 0.99(-7) 100.00(100.00) 0.99(-7) 100.00(100.00) 0.99(-7) 100.00(100.00) 0.95(-35) 1.11(1.51) 100.00(100.00) 0.57(-5) 1.11(1.51) 100.00(100.00) 0.57(-5) 1.18(-7) 100.00(100.00) 0.57(-5) 1.18(-7) 100.00(100.00) 0.57(-5) 1.18(-7) 100.00(100.00) 0.57(-5) 1.18(-7) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.00) 0.52(0.81) 1.48(1.50) 100.00(100.	100	SMOP6	0.82(0.82)*	1.25(4.98)*	100.00(100.00)=	0.82(0.82)*	1.25(5.47)*	100.00(100.00)=	0.82(0.81)*	1.25(6.94)*	100.00(100.00)=	0.82(0.82)*	1.25(15.81)*	100.00(100.00)=
$ \begin{array}{ $	100	SMOP7	0.34(0.32)*	0.94(4.46)*	100.00(100.00)=	0.34(0.35)	0.94(4.26)*	100.00(100.00)=	0.34(0.33)=	$0.94(5.73)^{+}$	100.00(100.00)=	0.34(0.33)*	0.94(12.94)*	$100.00(100.00)^{=}$
200 SMOP 0.580.57 1.21(5.19) 0.000(1000) 0.580.55 1.21(5.19) 0.000(1000) 200 SMOP2 0.560.54 1.12(5.19) 0.000(1000) 0.560.55 1.12(5.19) 0.000(1000) 200 SMOP2 0.570.55 1.12(5.19) 0.000(1000) 0.550.55 1.12(5.19) 0.000(1000) 200 SMOP3 0.570.55 1.12(5.19) 0.000(1000) 0.550.55 1.12(5.19) 0.000(1000) 200 SMOP4 0.520.81) 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.48(5.01) 0.000(1000) 0.520.821 1.77(1.242) 0.000(1000) 0.520.811 1.24(5.01) <th< td=""><td>100</td><td>SMOP8</td><td>0.19(0.16)*</td><td>0.99(6.10)*</td><td>100.00(77.50)*</td><td>0.19(0.18)=</td><td>0.99(4.17)*</td><td>100.00(100.00)=</td><td>0.19(0.15)*</td><td>0.99(7.53)*</td><td>100.00(100.00)=</td><td>0.19()*</td><td>0.99()*</td><td>100.00()*</td></th<>	100	SMOP8	0.19(0.16)*	0.99(6.10)*	100.00(77.50)*	0.19(0.18)=	0.99(4.17)*	100.00(100.00)=	0.19(0.15)*	0.99(7.53)*	100.00(100.00)=	0.19()*	0.99()*	100.00()*
200 SMOP2 0.05(0.59) 1.12(5.37) 10.00(10.007) 0.56(0.55) 1.12(3.59) 10.00(10.007) 200 SMOP4 0.57(0.56) 1.18(5.01) 10.00(10.000) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.18(5.7) 10.00(10.007) 0.57(0.55) 1.48(5.8) 10.00(10.007) 0.57(0.55) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(5.7) 10.00(10.007) 0.57(0.57) 1.48(7.7) 10.00(10.007) 0.57(0.57) 1.48(7.57) 10.00(10.007) 0.57(0.57) 1.48(7.57) 10.00(10.007) 0.57(0.57) 1.48(7.57) 10.00(10.007) 0.57(0.57)	200	SMOP1	0.58(0.57)*	1.21(5.05)*	100.00(100.00)=	$0.58(0.58)^{=}$	1.21(5.66)*	100.00(100.00)=	0.58(0.56)*	1.21(7.33)*	$100.00(100.00)^{=}$	0.58(0.57)*	1.21(15.19)*	$100.00(100.00)^{=}$
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	200	SMOP2	0.56(0.54)*	1.12(5.38)*	100.00(100.00)=	0.56(0.58)	1.12(5.63)*	100.00(100.00)=	0.56(0.54)*	1.12(6.77)*	100.00(100.00)=	0.56(0.55)+	1.12(13.90)*	100.00(100.00)=
200 SMOP# 8.28(82) 1.29(6.19) 100.0(10.00) 8.28(83) 1.29(6.5.82) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(83) 1.48(5.6.3) 100.00(10.00) 8.28(1.83) 1.48(1.81) 100.00(10.00) 0.23(0.81) 1.48(1.81) 100.00(10.00) 0.23(0.81) 1.48(1.81) 100.00(10.00) 0.23(0.81) 1.48(1.81) 100.00(10.00) 0.23(0.81) 1.48(1.81) 100.00(10.00) 0.23(0.81) 1.22(7.89) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.82(1.82) 100.00(10.00) 0.23(0.81) 1.12(1.83) 100.00(10.00)	200	SMOP3	0.57(0.56)*	1.18(5.01)*	100.00(100.00)=	0.57(0.58)	1.18(6.07)*	100.00(100.00)=	0.57(0.56)*	1.18(7.85)*	100.00(100.00)=	0.57()*	1.18()*	100.00()*
200 SMOPS 0.820.081 1.196(.50) 10.000(10.00) 0.820.821 1.196(.56) 10.000(10.00) 0.820.811 1.486.551 100.000(10.00) 0.820.811 1.486.551 100.000(10.00) 0.820.811 1.486.551 100.000(10.00) 0.820.811 1.487.51 100.000(10.00) 0.820.811 1.486.551 100.000(10.00) 0.820.811 1.487.51 100.000(10.00) 0.820.811 1.487.51 100.000(10.00) 0.820.811 1.487.51 100.000(10.00) 0.820.812 1.487.51 100.000(10.00) 0.820.812 1.487.51 100.000(10.00) 0.820.812 1.487.51 100.000(10.00) 0.820.812 1.597.51 100.000(10.00) 0.820.812 1.597.51 100.000(10.00) 0.820.812 1.597.51 100.000(10.00) 0.820.812 1.597.51 100.000(10.00) 0.820.812 1.597.55 100.000(10.00) 0.820.812 1.597.55 100.000(10.00) 0.820.811 1.787.81 100.000(10.00) 0.820.812 1.597.55 100.000(10.00) 0.820.812 1.597.55 100.000(10.00) 0.820.812 1.597.55 100.000(10.00) 0.820.811	200	SMOP4	0.82(0.82)	$1.20(4.80)^{\circ}$	100.00(100.00)=	0.82(0.82)	1.20(6.19)*	100.00(100.00)=	0.82(0.82)	1.20(11.39)*	100.00(100.00)	0.82(0.82)	1.20(26.82)*	100.00(100.00)=
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	200	SMOP5	0.82(0.81)*	1.19(6.10)*	100.00(100.00)=	0.82(0.82)*	1.19(6.50)*	100.00(100.00)=	0.82(0.81)+	1.19(13.68)+	100.00(100.00)=	0.82(0.82)+	1.19(21.46)*	100.00(100.00)=
200 SMOP? 0.130(23) 0.14(61.53) 0.000(100.00) 0.320(23) 0.14(14.51) 10.000(100.00) 0.320(23) 1.14(15.43) 10.000(100.00) 0.320(23) 1.14(15.43) 10.000(100.00) 0.320(23) 1.14(15.43) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.82(1.83) 10.000(100.00) 0.370(15) 1.92(1.54) 10.000(100.00) 0.320(15) 1.93(1.57) 10.000(100.00) 0.320(15) 1.93(1.57) 10.000(100.00) 0.320(15) 1.93(1.57) 10.000(100.00) 0.320(15) 1.93(1.57) 10.000(100.00) 0.320(15) 1.93(1.52) 10.000(100.00) 0.320(15) 1.93(1.57) 10.000(100.00) 0.320(1.51) 1.93(1.57) 10.000(100.00) 0.320(1.51) 1.93(1.57) 10.000(100.00) 0.320(1.51) 1.93(1.57) 10.000(100.00) 0.320(1.51) 1.93(1.57) <t< td=""><td>200</td><td>SMOP6</td><td>0.82(0.81)*</td><td>1.48(5.63)*</td><td>100.00(100.00)=</td><td>0.82(0.81)*</td><td>1.48(6.63)*</td><td>100.00(100.00)=</td><td>$0.82(0.81)^{+}$</td><td>$1.48(13.01)^{+}$</td><td>100.00(100.00)</td><td>$0.82(0.81)^{+}$</td><td>1.48(20.46)*</td><td>100.00(100.00)=</td></t<>	200	SMOP6	0.82(0.81)*	1.48(5.63)*	100.00(100.00)=	0.82(0.81)*	1.48(6.63)*	100.00(100.00)=	$0.82(0.81)^{+}$	$1.48(13.01)^{+}$	100.00(100.00)	$0.82(0.81)^{+}$	1.48(20.46)*	100.00(100.00)=
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	200	SMOP7	0.32(0.29)*	1.16(5.53)*	100.00(100.00)=	0.32(0.34)	1.16(5.89)*	100.00(100.00)=	0.32(0.29)*	$1.16(7.43)^{+}$	$100.00(100.00)^{=}$	$0.32(0.30)^{+}$	1.16(13.61)*	$100.00(100.00)^{=}$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	200	SMOP8	$0.17(0.14)^{+}$	1.19(5.49)*	100.00(93.00)*	0.17(0.15)*	1.19(6.23)*	100.00(100.00)*	0.17(0.11)*	$1.19(7.20)^{+}$	100.00(100.00)=	0.17()*	1.19(—)*	100.00()*
400 SMOP2 0.54(0.49) 1.77(1.83) 100.00(100.00) 0.54(0.51) 1.77(12.42) 100.00(100.00) 0.56(0.55) 1.66(2.59) 100.00(100.00) 0.56(0.55) 1.66(2.59) 100.00(100.00) 0.56(0.55) 1.66(2.59) 100.00(100.00) 0.52(0.52) 1.06(0.50) 1.06(0.50) 1.06(0.50) 1.06(0.50) 1.06(0.50) 1.06(0.50) 1.06(0.50) 1.06(0.50) 0.56(0.51) 1.06(0.50) 0.00(100.00) 0.52(0.51) 1.00(0.100.00) 0.52(0.51) 1.01(2.50) 1.00(0.58(0.51) 1.17(1.87) 100.00(100.00) 0.52(0.51) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.01(2.50) 1.02(0.50) 0.52(0.51) 1.88(1.80) 100.01(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.56(0.51) 1.17(1.75) 100.00(10.00) 0.55(0.51) 1.17(1.75) 100.00(10.00) <td>400</td> <td>SMOP1</td> <td>$0.57(0.55)^{+}$</td> <td>$1.82(6.87)^{*}$</td> <td>$100.00(100.00)^{=}$</td> <td>$0.57(0.57)^{*}$</td> <td>1.82(8.48)*</td> <td>$100.00(100.00)^{=}$</td> <td>0.57(0.55)*</td> <td>1.82(13.28)*</td> <td>$100.00(100.00)^{=}$</td> <td>$0.57(0.54)^{+}$</td> <td>1.82(17.89)*</td> <td>$100.00(100.00)^{=}$</td>	400	SMOP1	$0.57(0.55)^{+}$	$1.82(6.87)^{*}$	$100.00(100.00)^{=}$	$0.57(0.57)^{*}$	1.82(8.48)*	$100.00(100.00)^{=}$	0.57(0.55)*	1.82(13.28)*	$100.00(100.00)^{=}$	$0.57(0.54)^{+}$	1.82(17.89)*	$100.00(100.00)^{=}$
	400	SMOP2	0.54(0.49)*	1.77(7.38)*	100.00(100.00)=	0.54(0.54)=	1.77(8.87)*	100.00(100.00)=	0.54(0.51)*	1.77(12.42)+	100.00(100.00)=	0.54(0.49)*	1.77(18.30)+	100.00(100.00)=
400 SMOP4 0.82(0.82) 1.90(7.59) 100.00(10.000) 0.82(0.82) 1.90(2.56) 100.00(10.000) 0.82(0.82) 1.90(2.56) 100.00(10.000) 0.82(0.82) 1.90(2.56) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.81) 2.54(2.84) 100.00(10.000) 0.82(0.82) 1.77(7.87) 100.00(10.000) 0.51(0.41) 1.88(1.84) 100.00(10.000) 0.51(0.42) 1.28(2.85) 100.00(10.000) 0.51(0.42) 1.22(2.35) 100.00(10.000) 0.51(0.42) 1.22(2.45) 100.00(10.000) 0.51(0.42) 1.22(2.45) 100.00(10.000) 0.51(0.42) 1.22(2.45) 100.00(10.000) 0.51(0.42) 1.22(2.45) 100.00(10.000) 0.51(0.42) 1.22(2.45) 100.00(10.000) 1.22(2.45) 100.00(10.000)<	400	SMOP3	0.56(0.53)*	1.76(6.69)*	100.00(100.00)=	0.56(0.55)*	1.76(8.29)*	100.00(100.00)=	0.56(0.55)*	1.76(12.92)*	100.00(100.00)=	0.56()*	1.76(—)*	100.00()*
$ \begin{array}{c} 400 \\ 8MOP5 \\ 0.820(81) \\ 1.91(27.58) \\ 1.9000(100.00) \\ 0.820(81) \\ 2.54(-65) \\ 1.0000(100.00) \\ 0.820(81) \\ 2.54(-65) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.77(17.87) \\ 1.0000(100.00) \\ 0.280(31) \\ 1.17(17.87) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(52) \\ 1.12(13.67) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.580(51) \\ 1.0000(100.00) \\ 0.810(81) \\ 1.22(11.77) \\ 1.0000(100.00) \\ 0.810$	400	SMOP4	$0.82(0.82)^{=}$	1.90(5.90)*	100.00(100.00)=	0.82(0.82)	1.90(7.59)*	100.00(100.00)=	$0.82(0.82)^{+}$	1.90(22.65)*	$100.00(100.00)^{=}$	0.82(0.82)	1.90(31.57)*	$100.00(100.00)^{=}$
$ \begin{array}{c} 400 \\ 8MOF6 \\ 0.82(0.81) \\ 0.82(0.81$	400	SMOP5	0.82(0.81)*	1.91(7.28)*	100.00(100.00)=	0.82(0.82)*	1.91(9.13)*	100.00(100.00)=	$0.82(0.81)^{+}$	1.91(26.94)*	100.00(100.00)=	$0.82(0.81)^{+}$	1.91(27.95)*	100.00(100.00)=
$ \begin{array}{c} 400 \\ SMOP' \\ 0.28(0.24) \\ 1.77(1787) \\ 100.00(100.00) \\ 100.00(100.00) \\ 1.87(0.81) \\ 1.87(0.81) \\ 1.81(1.075) \\ 100.00(100.00) \\ 1.56(0.51) \\ 1.18(1.075) \\ 100.00(100.00) \\ 1.56(0.51) \\ 1.18(1.57) \\ 100.00(100.00) \\ 1.16(0.51) \\ 1.12(23.58) \\ 1.100.00(100.00) \\ 1.16(0.51) \\ 1.12(23.58) \\ 1.100.00(100.00) \\ 1.16(0.51) \\ 1.12(23.58) \\ 1.100.00(100.00) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.58) \\ 1.100.00(100.00) \\ 1.12(23.53) \\ 1.12(23.56) \\ 1.100.00(100.00) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.58) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.56(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.15(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.15(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.15(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.15(0.51) \\ 1.12(23.57) \\ 1.100.00(100.00) \\ 0.12(0.15) \\ 1.12(23.57) \\ 1.100.$	400	SMOP6	0.82(0.81)*	2.54(7.68)*	100.00(100.00)=	0.82(0.81)*	2.54(9.15)*	100.00(100.00)=	0.82(0.81)*	2.54(26.67)*	100.00(100.00)=	$0.82(0.81)^{+}$	2.54(28.49)*	100.00(100.00)=
$ \begin{array}{c} 400 \\ SM0PR \\ 60, 158(0.69) \\ 1, 188(-0.8) \\ 1, 188(-0.8) \\ 1, 100, 000, 00, 00, 00, 00, 00, 00, 00$	400	SMOP7	$0.28(0.24)^{+}$	1.77(7.87)*	$100.00(100.00)^{=}$	0.28(0.31)	$1.77(8.78)^*$	$100.00(100.00)^{=}$	$0.28(0.23)^{+}$	$1.77(12.73)^{+}$	$100.00(100.00)^{=}$	$0.28(0.23)^{+}$	$1.77(17.96)^{+}$	$100.00(100.00)^{=}$
800 SMOPI 0.56(0.51) 3.13(10.75) 100.00(100.00) 0.56(0.53) 3.13(24.36) 100.00(100.00) 0.56(0.51) 3.13(23.86) 100.00(100.00) 800 SMOP3 0.56(0.51) 3.12(1.97) 100.00(100.00) 0.51(0.47) 3.12(21.98) 100.00(100.00) 0.55(0.47) 3.12(2.18) 100.00(100.00) 0.55(0.47) 3.12(2.18) 100.00(100.00) 0.55(0.47) 3.12(2.18) 100.00(100.00) 0.55(0.47) 100.00(100.00) 0.55(0.47) 3.12(1.97) 100.00(100.00) 0.55(0.47) 3.12(2.18) 100.00(100.00) 0.55(0.47) 100.00(100.00) 0.52(0.82) 3.50(1.07) 100.00(100.00) 0.82(0.81) 3.29(1.52) 100.00(100.00) 0.82(0.81) 3.29(1.52) 100.00(100.00) 0.82(0.81) 3.29(1.52) 100.00(100.00) 0.82(0.81) 3.29(1.51) 100.00(100.00) 0.82(0.81) 3.22(1.52) 100.00(100.00) 0.82(0.81) 3.23(21.41) 100.00(100.00) 0.82(0.81) 3.23(21.41) 100.00(100.00) 0.12(0.01) 3.23(21.41) 100.00(100.00) 0.12(0.01) 3.23(21.41) 100.00(100.00) 0.12(0.01)	400	SMOP8	$0.15(0.09)^{+}$	1.88(7.08)*	100.00(99.00)*	0.15(0.11)*	$1.88(8.88)^{\circ}$	100.00(100.00)*	0.15(0.05)*	$1.88(11.80)^{+}$	$100.00(100.00)^{=}$	0.15()*	1.88()*	100.00()*
800 SMOP2 0.51(0.43) 3.12(11.10) 100.00(100.00) 0.51(0.43) 3.12(13.06) 100.00(100.00) 0.55(0.43) 100.00(100.00) 0.51(0.44) 3.12(23.45) 100.00(100.00) 800 SMOP4 0.82(0.82) 3.50(1.47) 100.00(100.00) 0.82(0.82) 3.50(1.47) 100.00(100.00) 0.82(0.81) 3.29(1.40) 100.00(100.00) 0.82(0.81) 3.29(4.23) 100.00(100.00) 800 SMOP6 0.82(0.79) 4.54(1.12) 100.00(100.00) 0.82(0.81) 3.29(1.40) 100.00(100.00) 0.82(0.81) 4.54(39.07) 100.00(100.00) 800 SMOP6 0.82(0.79) 4.54(1.12) 100.00(100.00) 0.24(0.01) 3.12(12.91) 100.00(100.00) 0.82(0.81) 4.54(2.30) 100.00(100.00) 800 SMOP7 0.24(0.43) 5.31(8.17) 100.00(100.00) 0.24(0.43) 5.74(2.83) 100.00(100.00) 0.12(0.1) 3.23(2.1.3) 100.00(100.00) 0.12(0.1) 3.23(2.1.3) 100.00(100.00) 0.12(0.1) 3.23(2.2.3) 100.00(100.00) 0.12(0.1) 3.23(2.1.3) 100.00(100.00)	800	SMOP1	0.56(0.51)*	3.13(10.75)*	100.00(100.00)=	0.56(0.52)*	3.13(11.97)*	100.00(100.00)=	0.56(0.53)*	3.13(24.36)*	$100.00(100.00)^{=}$	$0.56(0.51)^{+}$	3.13(23.86)*	$100.00(100.00)^{=}$
800 SMOP3 0.56(0.51) 3.19(10.11) 100.00(10.00) 0.56(0.51) 3.19(13.65) 100.00(100.00) 0.82(0.82) 3.50(1.84) 100.00(100.00) 0.82(0.82) 3.50(1.84) 3.29(1.42) 100.00(100.00) 0.82(0.82) 3.50(1.84) 3.29(1.42) 100.00(100.00) 0.82(0.82) 3.50(1.84) 3.29(1.42) 100.00(100.00) 0.82(0.82) 3.50(1.84) 3.29(1.42) 100.00(100.00) 0.82(0.81) 4.54(7.32) 100.00(100.00) 0.82(0.81) 4.54(7.32) 100.00(100.00) 0.24(0.15) 3.15(1.17) 100.00(100.00) 0.24(0.15) 3.15(1.17) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.24(0.16) 3.15(2.52) 100.00(100.00) 0.56(0.50) 5.35(2.57) 100.00(100.00) 0.56(0.50) 5.35(2.57) 100.00(100.00) 0.55(0.49)	800	SMOP2	0.51(0.43)*	3.12(11.10)*	100.00(100.00)=	0.51(0.45)*	3.12(13.06)*	100.00(100.00)=	0.51(0.47)*	3.12(19.98)*	100.00(100.00)=	$0.51(0.44)^{+}$	3.12(23.45)*	100.00(100.00)=
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	800	SMOP3	0.56(0.50)*	3.19(10.11)*	100.00(100.00)=	0.56(0.51)*	3.19(13.65)*	100.00(100.00)=	0.56(0.53)*	3.19(20.05)*	100.00(100.00)=	0.56()*	3.19()*	100.00()*
$ \begin{array}{c} 800 \mathrm{SMOP5} & 0.81(0.80) & 3.29(11.42) & 100.00(10.00) & 0.81(0.81) & 3.29(11.50) & 100.00(100.00) & 0.81(0.81) & 3.29(11.50) & 100.00(100.00) \\ 800 \mathrm{SMOP7} & 0.24(0.15) & 3.15(11.17) & 100.00(100.00) & 0.24(0.80) & 4.54(14.06) & 100.00(100.00) & 0.24(0.20) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.24(0.16) & 3.15(25.22) & 100.00(100.00) & 0.12(-) & 3.23(-) & 100.00(100.00) & 1.2(-) & 3.23(-) & 100.00(100.00) & 0.56(0.50) & 5.53(46.58) & 100.00(100.00) & 0.55(0.49) & 5.53(25.23) & 100.00(100.00) & 0.54(0.42) & 5.70(3.81) & 100.00(100.00) & 0.55(0.49) & 5.89(2.399) & 100.00(100.00) & 0.55(0.49) & 5.89(-) & 100.00(100.00) & 0.55(-) & 5.89(-) & 100.00(100.00) & 0.55(0.49) & 5.89(2.399) & 100.00(100.00) & 0.81(0.81) & 5.92(5.77) & 100.00(100.00) & 0.55(0.49) & 5.89(2.39) & 100.00(100.00) & 0.81(0.81) & 5.92(5.587) & 100.00(100.00) & 0.55(0.49) & 5.82(2.587) & 100.00(100.00) & 0.81(0.81) & 5.92(5.587) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.57) & 100.00(100.00) & 0.81(0.81) & 5.92(5.7) & 100.00(100.00) & 0.81(0.81) & 5.92(5.7) & 100.00(100.00) & 0.81(0.81) & 5.92(5.7) & 100.00(100.00) & 0.81(0.81) & 5.92(5.7) & 100.00(1$	800	SMOP4	$0.82(0.82)^{=}$	3.50(7.98)*	$100.00(100.00)^{=}$	$0.82(0.82)^{=}$	3.50(11.47)*	$100.00(100.00)^{=}$	$0.82(0.82)^{+}$	3.50(51.07)*	$100.00(100.00)^{=}$	0.82(0.82)	3.50(46.23)*	$100.00(100.00)^{=}$
$ \begin{array}{c} 800 \\ 800 $	800	SMOP5	$0.81(0.80)^{+}$	3.29(11.42)*	$100.00(100.00)^{=}$	$0.81(0.81)^{=}$	3.29(13.52)*	100.00(100.00)=	$0.81(0.81)^{+}$	3.29(71.96)*	100.00(100.00)=	$0.81(0.81)^{+}$	3.29(41.23)*	$100.00(100.00)^{=}$
$ \begin{array}{c} 800 \\ 800 Pr $	800	SMOP6	$0.82(0.79)^{+}$	4.54(11.72)*	$100.00(100.00)^{=}$	0.82(0.80)*	4.54(14.06)*	$100.00(100.00)^{=}$	$0.82(0.80)^{+}$	4.54(72.83)*	$100.00(100.00)^{=}$	$0.82(0.81)^{+}$	4.54(39.07)*	$100.00(100.00)^{=}$
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	800	SMOP7	0.24(0.15)*	3.15(11.17)*	100.00(100.00)-	0.24(0.19)*	3.15(13.63)*	100.00(100.00)	0.24(0.20)*	3.15(25.22)*	100.00(100.00)	0.24(0.16)*	3.15(22.47)*	100.00(100.00)
	800	SMOP8	0.12(0.04)*	3.23(11.77)*	100.00(100.00)*	0.12(0.05)*	3.23(12.61)*	100.00(100.00)*	0.12(0.01)*	3.23(21.34)*	100.00(100.00)-	0.12()*	3.23(—)*	100.00()*
	1600	SMOP1	0.56(0.49)*	5.53(18.86)*	100.00(100.00)-	0.56(0.49)*	5.53(22.63)*	100.00(100.00)	$0.56(0.50)^{+}$	5.53(46.58)*	100.00(100.00)-	$0.56(0.50)^{+}$	5.53(29.52)*	100.00(100.00)-
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1600	SMOP2	0.48(0.39)*	5.70(20.88)*	100.00(100.00)-	0.48(0.40)*	5.70(23.98)*	100.00(100.00)-	0.48(0.42)*	5.70(45.15)*	100.00(100.00)-	$0.48(0.41)^{+}$	5.70(30.41)*	100.00(100.00)-
$ 1600 SMOP4 \\ 0.82(0.82) 6.42(11.32) 100.00(100.00) \\ 0.82(0.82) 6.42(17.15) \\ 100.00(100.00) \\ 1000 SMOP5 \\ 0.81(0.78) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.78) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.78) 5.52(15.57) \\ 100.00(100.00) \\ 100.00) \\ 1000 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.81(0.81) 5.52(15.57) \\ 100.00(100.00) \\ 0.18(0.12) 5.52(15.57) \\ 100.00(100.00) \\ 0.18(0.12) 5.52(15.57) \\ 100.00(100.00) \\ 0.55(0.49) 11.00(57.09) \\ 11.00(57.09) 100.00(100.00) \\ 0.55(0.49) 11.00(57.09) \\ 11.00(57.09) 100.00(100.00) \\ 0.55(0.49) 11.50(74.88) \\ 100.00(100.00) 0.55(0.49) 11.00(57.09) \\ 11.50(15.03) 11.50(74.88) \\ 100.00(100.00) 0.55(0.49) 11.00(57.09) \\ 0.46(0.39) 11.59(74.33) \\ 100.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.28(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.28(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 10.00(100.00) 0.54(0.49) 11.88(7.50) \\ 11.88(7.50) 0.54(0.78) 12.24(7.3.8) 100.00(100.00) \\ 0.54(0.49) 12.44(7.3.8) 100.00(100.00) \\ 0.54(0.49) 0.54(0.49) 12.24(7.3.8) 100.00(100.00) \\ 0.54(0.49) 0.54(0.49) 0.54(0.57) 12.24(7.58) 100.00(100.00) \\ 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) 0.54(0.49) $	1600	SMOP3	0.55(0.49)*	5.89(21.58)*	100.00(100.00)-	0.55(0.49)*	5.89(23.99)*	100.00(100.00)-	0.55(0.50)*	5.89(40.62)*	100.00(100.00)-	0.55(—)*	5.89(—)*	100.00()*
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1600	SMOP4	0.82(0.82)	6.42(11.32)	$100.00(100.00)^{-1}$	0.82(0.82)	6.42(15.15)*	100.00(100.00)	0.82(0.82)*	6.42(110.29)	100.00(100.00)	0.82(0.82)	6.42(77.75)	100.00(100.00)
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1600	SMOP5	$0.81(0.78)^{\circ}$	5.92(20.43)*	100.00(100.00)	0.81(0.79)*	5.92(23.74)*	100.00(100.00)	0.81(0.81)*	5.92(155.87)*	100.00(100.00)	0.81(0.81)	5.92(52.77)*	100.00(100.00)
	1600	SMOP6	0.81(0.78)	8.57(23.15)	100.00(100.00)	0.81(0.78)	8.57(25.00)	100.00(100.00)	0.81(0.80)	8.57(100.09)	100.00(100.00)	0.81(0.80)	8.57(58.09)	100.00(100.00)
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1600	SMOP/	0.18(0.11)	5.72(21.56)	100.00(100.00)	0.18(0.12)	5.72(24.57)	100.00(100.00)	0.18(0.15)	5.72(43.85)	100.00(100.00)	0.18(0.12)	5.72(29.95)	100.00(100.00)
$ \begin{array}{c} 3200 \text{SMOP1} \\ 3200 \text{SMOP2} \\ 0.450(.49) 11.000, (0.00) \\ 0.50(.49) 11.000, (0.00) \\ 0.50(.49) 11.000, (0.00) \\ 0.50(.49) 11.000, (0.00) \\ 0.50(.49) 11.000, (0.00) \\ 0.50(.49) 11.000, (0.00) \\ 0.50(.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.88(7.50) \\ 100.00(100.00) \\ 0.54(0.49) 11.24(73.78) \\ 100.00(100.00) \\ 0.80(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.24(73.78) \\ 100.00(100.00) \\ 0.81(0.80) 12.16(8.39) \\ 100.00(100.00) \\ 0.81(0.80) 12.16(8.39) \\ 100.00(100.00) \\ 0.54(0.47) 25.59(4.57) \\ 100.00(100.00) \\ 0.54(0.47) 25.59(4.57) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.99) \\ 100.00(100.00) \\ 0.54(0.49) 25.59(12.90) \\ 0.54(0.49) 25.59(12.90) \\ 0.54(0.49) 25.59(12.91) \\ 0.00(100.0$	1600	SMOP8	0.10(0.01)	5.94(19.50)	100.00(100.00)	0.10(0.01)	5.94(24.51)	100.00(100.00)	0.10(0.00)*	5.94(57.09)	100.00(100.00)	0.10()*	5.94()	100.00()*
$ \begin{array}{c} 5200 \text{SMOP2} \\ 3200 \text{SMOP3} \\ 0.54(0.39) 11.59(4.3.3) 100.00(100.00) \\ 0.000(100.00) \\ 0.54(0.49) \\ 11.88(75.87) \\ 100.00(100.00) \\ 0.54(0.49) \\ 11.88(75.87) \\ 100.00(100.00) \\ 0.54(0.49) \\ 11.88(75.87) \\ 100.00(100.00) \\ 0.52(0.42) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.82(0.82) \\ 13.10(27.81) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.24(32.72) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.24(32.72) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.24(32.72) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.24(32.72) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.24(32.81) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.16(64.33) \\ 100.00(100.00) \\ 0.81(0.80) \\ 12.16(64.33) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.87) \\ 12.26(149.06) \\ 100.00(100.00) \\ 0.81(0.80) \\ 0.81(0.80) \\ 0.81(0.80) \\ 0.81(0.80) \\ 0.$	3200	SMOPI	0.55(0.49)	11.00(70.05)	100.00(100.00)	0.55(0.49)	11.00(69.80)	100.00(100.00)	0.55(0.49)	11.00(95.77)	100.00(100.00)	0.55(0.49)	11.00(45.52)	100.00(100.00)
$ \begin{array}{c} 5200 \text{SMOP} \\ 5200 \text{SMOP} $	3200	SMOP2	0.46(0.38)*	11.59(74.33)	$100.00(100.00)^{-1}$	0.46(0.38)	11.59(74.88)	100.00(100.00)	0.46(0.39)*	11.59(84.90)*	100.00(100.00)	0.46(0.39)	11.59(45.88)*	100.00(100.00)-
$ \begin{array}{c} 5200 \text{SMOF4} \\ 0.52(0.52) 15.10(15.58) 100.00(10.000) \\ 0.52(0.52) 15.10(21.24) \\ 100.00(10.000) \\ 1200 \text{SMOF5} \\ 0.58(0.78) 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.26(149.06) \\ 12.24(73.78) \\ 100.00(10.000) \\ 12.16(138.93) \\ 100.00(100.00) \\ 12.16(138.93) \\ 1$	3200	SMOP3	0.54(0.49)	11.88(72.50)	100.00(100.00)	0.54(0.49)	11.88(75.07)	100.00(100.00)	0.54(0.49)	11.88(77.84)	100.00(100.00)	0.54()	11.88()	100.00()
$ \begin{array}{c} 5200 \\ 5007\\ 3200 \\ 3200 \\ 5007\\ 3200 \\ 5007\\ 5000\\ 500000\\ 50000\\ 50000\\ 50000\\ 50000\\ 500000\\ 50000\\ 50000\\ 50000\\ 500000\\ 500000\\ 500000\\ 500000\\ 50000\\ 50000\\ 50000\\ 50000\\ 50000\\ 50000\\ 500000\\$	3200	SMOP4	0.82(0.82)	13.10(15.58)	100.00(100.00)	0.82(0.82)	13.10(20.21)	100.00(100.00)	0.82(0.82)	13.10(270.81)	100.00(100.00)	0.82(0.82)	13.10(171.40) 12.24(72.78)	100.00(100.00)
$ \begin{array}{c} 5200 \text{SMOPF} \\ 3200 \text{SMOPF} \\ 0.510(.85) \\ 1.529(347) \\ 1.59(347$	3200	SMOPS	0.80(0.78)	12.24(71.43)	$100.00(100.00)^{-1}$	0.80(0.78)	12.24(75.76)	$100.00(100.00)^{-1}$	0.81(0.80)*	12.24(325.72)	100.00(100.00)	0.80(0.80)	12.24(75.76)	100.00(100.00)
$ \begin{array}{c} \frac{5200}{6400} & \frac{5MOP}{6400} & \frac{12,16(633)}{63000} & \frac{12,16(633)}{10000} & \frac{12,16(643)}{100000} & \frac{12,16(643)}{1000000} & \frac{12,16(138,33)}{100000(10000)} & \frac{12,16(138,33)}{10000(10000)} & \frac{12,16(138,33)}{10000(100$	2200	SMOP0	0.81(0.78)	17.24(79.92)	$100.00(100.00)^{-1}$	0.81(0.78) $0.15(0.10)^{+}$	17.24(79.06)	$100.00(100.00)^{-1}$	0.81(0.80)	17.24(406.72)	$100.00(100.00)^{-1}$	0.81(0.79)	17.24(00.03)	100.00(100.00)
$ \begin{array}{c} \frac{1}{2600} & \frac{5}{3007} & \frac{1}{24.702} & \frac{1}{24.100} & $	3200	SMOP/	0.13(0.10)	12.00(79.92)	100.00(100.00)	0.13(0.10)	12.00(60.78)	100.00(100.00)	0.13(0.09)	12.00(149.00)	100.00(100.00)	0.15(0.10)	12.00(47.99)	100.00(100.00)
$ \begin{array}{c} 6400 \text{SMOP1} \\ 6400 \text{SMOP2} \\ 0.440.37) 2.4.72(94.398) 100.00(9.00) \\ 0.580(0.7) 2.4.72(94.398) \\ 100.00(1.00) \\ 0.580(0.7) 2.4.72(94.398) \\ 100.00(1.00) \\ 0.580(0.7$	5200	SMORE	0.07(0.00)	24.72(244.08)*	100.00(100.00)	0.07(0.00)	12.10(00.45)	100.00(100.00)	0.07(0.00)	12.10(138.93)	100.00(100.00)	0.07()	24.72(70.90)	100.00()
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	6400	SMOPT	0.34(0.37)	5 50(347.67)	100.00(9.00)	0.34(0.37)	24.72(345.89)	100.00(9.00)	0.34(0.48)	24.72(210.79)	100.00(100.00)	0.34(0.49)	24.72(79.80)	100.00(100.00)
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	6400	SMOP2 SMOP2	$0.44(0.27)^{\circ}$ 0.53(0.35) ⁺	26.24(358.55)*	100.00(8.00)*	0.53(0.36)*	25.59(540.57)	100.00(8.00)*	0.53(0.49)*	26.24(180.98)*	100.00(100.00)	0.53()*	25.59(81.08)	100.00(100.00)
$ \begin{array}{c} 0.0007 \\ 6400 \\ 5MOP5 \\ 0.790(.68) \\ 26.39(66.2.41) \\ 100.001(1.00) \\ 100.001(1.00) \\ 0.790(.68) \\ 26.39(362.41) \\ 100.001(1.00) \\ 100.001(1.00) \\ 0.790(.68) \\ 26.39(355.57) \\ 100.001(1.00) \\ 100.001(1.00) \\ 0.790(.81) \\ 26.39(763.41) \\ 100.001(100.00) \\ 0.80(0.70) \\ 36.35(92.28) \\ 100.001(1.00) \\ 100.001(1.00) \\ 0.80(0.70) \\ 36.35(92.28) \\ 100.001(1.00) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 2.60(24.71,5) \\ 100.001(1.00) \\ 0.80(0.80) \\ 0.14(0.10) \\ 0.80$	6400	SMOP/	0.82(0.75)*	28.24(358.55)	100.00(7.00)	0.82(0.73)*	28.29(44.96)*	100.00(7.00)*	0.82(0.82)*	28.29(631.41)*	100.00(100.00)	0.82(0.82)	28.29(669.27)*	100.00(100.00)=
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	6400	SMOP5	0.82(0.73)	26.29(45.88)	100.00(11.00)*	0.79(0.66)*	26.29(44.90)	100.00(12.00)*	0.79(0.81)	26.29(051.41)	100.00(100.00)	0.79(0.79)*	26.39(118.16)*	100.00(100.00)
6400 SMOP7 0.14(0.04) 26.06(372.07) 100.00(7.00) 0.14(0.05) 26.06(370.04) 100.00(8.00) 0.14(0.10) 26.06(248.75) 100.00(100.00) 0.14(0.10) 26.06(370.04) 100.00(7.00) 0.14(0.10) 26.06(370.04) 100.00(7.00) 0.14(0.10) 26.06(248.75) 100.00(100.00) 0.14(0.10) 26.06(370.04) 100.00(7.00) 0.14(0.10) 26.06(248.75) 100.00(100.00) 0.05(0.00) 0.26.29(378.64) 100.00(7.00) 0.14(0.10) 26.06(248.75) 100.00(100.00) 0.05(0.00) 0.14(0.10) 26.06(370.04) 100.00(7.00) 0.14(0.10) 26.06(248.75) 100.00(100.00) 0.05(0.00) 0.05(0.00) 0.05(0.00) 26.29(378.64) 100.00(100.00) 0.05(0.00) 0.05(0.00) 26.29(378.64) 100.00(100.00) 0.05(0.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 26.29(271.75) 100.00(100.00) 0.05(0.00) 2	6400	SMOP6	0.80(0.70)*	36 35(392 58)*	100.00(16.00)*	0.80(0.70)*	36 35(402 79)*	100.00(12.00)*	0.80(0.80)-	36 35(861 83)*	100.00(100.00)=	0.80(0.78)*	36 35(174 14)*	100.00(100.00)
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	6400	SMOP7	$0.14(0.04)^{+}$	26.06(372.07)*	100.00(7.00)*	0.14(0.05)*	26.06(370.04)*	100.00(10.00)*	$0.14(0.10)^{+}$	26 06(248 75)*	100.00(100.00)=	0.14(0.10)*	26.06(97.86)*	$100.00(100.00)^{-1}$
$(+/-\infty)$ 500/4 5600 120/44 38/9 5600 110/45 49/3 5600 00/56 497/0 42/1/40 14/0/2	6400	SMOP8	$0.05(0.00)^{+}$	26 29(364 05)*	100.00(7.00)*	0.05(0.00)*	26 29(337 60)*	100.00(7.00)*	0.05(0.00)*	26 29(271 75)*	100.00(100.00)=	0.05()*	26 29()*	100.00(
		+/-/~)	50/2/4	56/0/0	12/0/44	38/9/9	56/0/0	11/0/45	49/4/3	56/0/0	0/0/56	49/7/0	42/14/0	14/0/42

Table 9: Performance of different algorithms compared to S-NSGA-II in solving two real-world problems: neural network training and portfolio optimization. Performance is measured in hypervolume, run time, and number of nondominated solutions. The performance of S-NSGA-II is the first number in each cell, followed by the base algorithm in parentheses. Scenarios where sparse NSGA-II performed better, worse, or approximately the same as the base method are denoted with +, -, and ≈ respectively. Scenarios where the base method did not halt within a reasonable amount of time are denoted with —.

Prob	Data	#	D	5	S-NSGA-II vs Spar	rseEA	S	-NSGA-II vs Spars	seEA2	S-	NSGA-II vs MOE	A/PSL	S-NSGA-II vs PM-MOEA			
	set	Layers		HV	Time	NDS	HV	Time	NDS	HV	Time	NDS	HV	Time	NDS	
NN	1	20	321	$0.89(0.89)^{=}$	30.36(35.44)*	$100.00(8.00)^{+}$	0.89(0.89)=	30.36(35.49)*	100.00(7.00)*	0.89(0.89)	30.36(37.80)*	$100.00(8.00)^{+}$	0.89(0.88)*	30.36(46.60)*	$100.00(100.00)^{=}$	
NN	1	40	641	$0.89(0.89)^{=}$	41.63(44.30)*	100.00(7.00)*	$0.89(0.89)^{=}$	41.63(46.02)*	100.00(6.50)*	0.89(0.90)	41.63(51.54)*	100.00(9.50)*	0.89(0.89)*	41.63(58.85)*	100.00(100.00)=	
NN	1	80	1281	$0.89(0.89)^{*}$	63.61(65.29)*	100.00(6.00)*	0.89(0.89)*	63.61(67.86) ⁺	100.00(6.00)*	$0.89(0.90)^{=}$	63.61(93.25)*	$100.00(8.00)^{+}$	0.89(0.89)*	63.61(88.72)*	100.00(100.00)=	
NN	1	160	2561	0.90(0.89)*	142.51(144.27)=	100.00(6.00)*	0.90(0.89)*	142.51(152.93)*	100.00(6.00)*	0.90(0.89)=	142.51(212.50)+	100.00(9.00)*	0.90(0.89)*	142.51(180.97)*	100.00(100.00)=	
NN	1	320	5121	$0.90(0.88)^{+}$	232.09(334.26)*	100.00(3.00)*	0.90(0.88)*	232.09(343.01)*	100.00(3.00)*	0.90(0.89)*	232.09(446.00)*	100.00(7.50)*	0.90(0.89)*	232.09(308.71)*	100.00(100.00)=	
NN	2	20	401	0.94(0.97)	24.02(33.24)*	100.00(12.00)*	0.94(0.97)	24.02(31.27)*	100.00(11.00)*	0.94(0.96)	24.02(36.59)*	100.00(6.00)*	0.94(0.93)=	24.02(45.94)*	100.00(100.00)=	
NN	2	40	801	0.95(0.98)	32.76(41.07)*	100.00(12.50)*	0.95(0.98)	32.76(44.43)*	100.00(14.00)*	0.95(0.95)=	32.76(54.80)*	100.00(6.00)*	0.95(0.93)*	32.76(62.30)*	$100.00(100.00)^{=}$	
NN	2	80	1601	0.96(0.98)-	51.58(61.31)+	100.00(13.00)*	0.96(0.98)	51.58(64.13)*	100.00(13.00)*	0.96(0.94)*	51.58(93.50)*	100.00(6.00)+	0.96(0.93)*	51.58(97.70)*	100.00(100.00)=	
NN	2	160	3201	0.96(0.97)-	108.65(117.14)*	100.00(9.00)*	0.96(0.97)-	108.65(125.12)*	100.00(9.00)*	0.96(0.97)=	108.65(232.33)*	100.00(7.50)*	0.96(0.94)*	108.65(187.73)*	100.00(100.00)=	
NN	2	320	6401	0.97(0.94)*	203.70(336.55)*	100.00(3.50)*	0.97(0.94)*	203.70(343.52)*	100.00(4.00)*	0.97(0.97)=	203.70(549.06)*	$100.00(8.00)^{+}$	0.97(0.94)*	203.70(332.14)*	$100.00(100.00)^{=}$	
NN	3	20	521	0.80(0.81)-	61.73(55.89)=	100.00(17.00)*	0.80(0.81)	61.73(49.04)-	100.00(16.00)*	0.80(0.80)=	61.73(64.72)=	100.00(11.00)*	0.80(0.79)*	61.73(97.19)*	100.00(100.00)=	
NN	3	40	1041	$0.80(0.81)^{-1}$	84.16(96.56)*	100.00(16.00)*	0.80(0.81)	84.16(98.37)*	100.00(15.50)*	$0.80(0.80)^{=}$	84.16(131.22)*	$100.00(9.00)^{+}$	$0.80(0.79)^{+}$	84.16(124.37)*	$100.00(100.00)^{=}$	
NN	3	80	2081	0.80(0.81)	121.86(124.04)=	100.00(19.00)*	0.80(0.81)	121.86(139.71)*	100.00(16.00)*	0.80(0.81)*	121.86(182.40)*	100.00(6.50)*	0.80(0.79)*	121.86(160.69)*	100.00(100.00)=	
NN	3	160	4161	0.80(0.78)*	213.53(221.27)=	100.00(4.00)*	0.80(0.78)*	213.53(231.61)=	100.00(4.00)*	0.80(0.81)=	213.53(360.23)*	100.00(5.50)*	0.80(0.79)*	213.53(314.69)*	100.00(100.00)=	
NN	3	320	8321	$0.81(0.78)^{+}$	462.53(967.34)*	$100.00(4.00)^{+}$	$0.81(0.78)^{+}$	462.53(1014.64)*	$100.00(4.00)^{+}$	$0.81(0.81)^{=}$	462.53(777.91)*	100.00(7.50)*	$0.81(0.80)^{+}$	462.53(571.96)*	$100.00(100.00)^{=}$	
PO	1	20	321	$0.12(0.12)^{=}$	9.69(30.78)*	100.00(100.00)=	0.12(0.12)	9.69(28.81)*	100.00(100.00)=	0.12(0.12)*	9.69(119.43)*	100.00(100.00)*	$0.12(0.12)^{=}$	9.69(203.27)*	100.00(100.00)=	
PO	2	40	641	$0.12(0.12)^{+}$	333.67(975.89)*	100.00(55.50)*	0.12(0.12)*	333.67(834.32)*	100.00(56.00)*	0.12(0.11)*	333.67(1146.78)*	100.00(100.00)*	0.12(0.12)=	33.67(6235.00)*	100.00(100.00)*	
	(+	/−/≈)		7/7/3	13/0/4	16/0/1	7/8/2	15/1/1	16/0/1	5/3/9	16/0/1	17/0/0	14/0/3	17/0/0	1/0/16	

Overall, we are able to reject the majority (34 out of 56) of HV null hypotheses $(H_0^{HV}:100/\text{SMOP}[5-6], H_0^{HV}:200/\text{SMOP}[5-6,8], H_0^{HV}:400/\text{SMOP}[1,3,5-6,8], H_0^{HV}:800/\text{SMOP}[1-3,6-8], H_0^{HV}:1600/\text{SMOP}[1-3,5-8], H_0^{HV}:3200/\text{SMOP}[1-3,6-8], and <math>H_0^{HV}:6400/\text{SMOP}[1-3,78]$). The remaining null hypotheses we accept. Other than deficient HVs for this subclass of problems, S-NSGA-II runs faster than SparseEA under all scenarios (Table 8), rejecting $H_0^{RT}:*/\text{SMOP}*$. S-NSGA-II also maintains more NDS than SparseEA under all test problems with 6,400 decision variables, and for all SMOP8 under 100-800 decision variables (Table 8). In fact, S-NSGA-II maintains a median 100 NDS in all scenarios, which is encouraging, considering many of the sparse large-scale optimization algorithms struggle to maintain 100 NDS, including SparseEA, SparseEA2, and MP-MMEA. Therefore, we reject all of $H_0^{NDS}:*/\text{SMOP}*$.

6.4.2 Real-world problem results

On the whole, S-NSGA-II excelled at solving the NN problem set in the same categories as it did in the SMOP problem set. Across the four problem sets, S-NSGA-II outperformed or kept pace with the competing algorithms in scenarios with large numbers of decision variables (i.e., approximately greater than 3,200 decision variables) (Table 9, Figure 19). Also, except for one scenario (NN, data set 3, vs. SparseEA2), S-NSGA-II ran as fast (six scenarios), if not faster (54 scenarios), than the competing algorithms (Table 9). We can therefore reject H_0^{RT} :*/NN_D*,



Figure 19: The studied algorithms training a NN with increasing complexity. Note that similar to many scenarios, SparseEA and SparseEA2 perform very well in lower decision spaces, while dropping in performance at around 3,000 decision variables. S-NSGA-II actually gradually performs better and better with increasing complexity, as do the other competing algorithms. The shaded areas around each mean line is the 95% confidence interval for the mean.

except for the aforementioned exceptions. Finally, S-NSGA-II maintained more NDS than all of SparseEA, SparseEA2, and MOEA/PSL, and approximately the same number of NDS as PM-MOEA (Table 9). In summary, we reject H_0^{NDS} :*/NN_D*.

Though S-NSGA-II performed well with respect to HV, it did not outperform the other algorithms as it did for the SMOP problems. As a matter of fact, we can only reject H_0^{HV} :320/NN_D1 (Table 9). For the rest of the scenarios, we accept the null hypotheses H_0^{HV} :*/NN_D*. However, the actual results are less simple than the HV hypothesis results suggest. First off, S-NSGA-II performed, with respects to HV, either better than or as well as all of the competing algorithms after 3,200 decision variables. In other words, no algorithm out competed S-NSGA-II in this problem set with respects to HV. For further context, consider that SNSGA-II maintains dramatically better NDS than SparseEA, SparseEA2, and MOEA/PSL (maximum median NDS of 100 versus 19) (Table 9). For these algorithms, an algorithm that achieves the same HV with improved NDS is a better outcome. The only other algorithms that achieved perfect median NDS for the NN problems was PM-MOEA, but S-NSGA-II achieves better HV than PM-MOEA for all scenarios with greater than 3,200 decision variables (Table 9). Therefore, despite accepting most of the HV hypotheses, the time and NDS improvements brought about by S-NSGA-II is arguably a worthwhile trade-off for sparse LSMOPs with greater than 3,200 decision variables.

The PO problem set yielded similar results to the NN problem set. First off, S-NSGA-II ran faster than all algorithms, thus rejecting H_0^{RT} :PO_D* (Table 9). Additionally, S-NSGAII achieved median NDS as high or higher than all other algorithms, therefore also rejecting H_0^{NDS} :PO_D* (Table 9). On the other hand, we must accept H_0^{HV} :PO_D*, since SNSGA-II does not universally increase HV as it does for the SMOP problem set (Table 9). Specifically,

SparseEA2 outperforms S-NSGA-II for PO data set 1 (1,000 decision variables), and PM-MOEA performs as well as S-NSGA-II for data set 2 (5,000 decision variables). However, similar tradeoffs between SparseEA, SparseEA2, and MOEA/PSL vs. SNSGA-II are still present: S-NSGA-II gets far better NDS and run time than the competing algorithms. PM-MOEA, on the other hand, gets statistically similar HV and NDS, but still runs slower than S-NSGA-II. The improved run time compared to PM-MOEA is arguably worthwhile for a user.

6.4.3 Discussion

This study recommends SparseEA2 for sparse LSMOPs with relatively lower numbers of decision variables, and SNSGA-II for problems with relatively higher numbers of decision variables. For the SMOP problem set, between 100 and 200 decision variables, SparseEA2 obtains the best HV for the majority of scenarios and maintains perfect median NDS (though SparseEA2 still runs universally slower than S-NSGA-II). However, when the number of decision variables increases (approximately 3,000 decision variables), SparseEA2, and its predecessor SparseEA, struggles to obtain good HVs (Figure 18, Figure A14—Figure A21) compared to the other algorithms. Also, for problems larger than 6,400 decision variables, SparseEA and SparseEA2 both fail to maintain an adequate number of NDS. During this data range is where S-NSGA-II performs the best. It achieves the best HV for the majority of scenarios, maintains perfect NDS, and runs universally faster than the competing algorithms.

The real-world problem sets also roughly uphold our recommendations. SparseEA2 performed the best with respects to HV below and at 3,200 decision variables. S-NSGA-II, though it did not obtain universally better HV, it maintained median HV values that were as good as any other algorithm while maintaining a stable NDS and running faster in the majority of scenarios. Therefore, though the range is different between the real world problems and the SMOP problems,

SparseEA2 performs better in lower ranges of decision variables, whereas S-NSGA-II performs better in higher ranges of decision variables. This upholds our general recommendation, but at a different scale of decision variables (i.e., > 200 versus > 3,200).

6.5 Conclusion

Through an extensive set of benchmark and real-world tests, we have determined that S-NSGA-II is a successful sparse LSMOP algorithm. With respect to convergence and diversity (i.e., hypervolume (HV) metric), S-NSGA-II either performs as well if not better than other state-of-the-art sparse LSMOP algorithms, particularly with problems with 5,000 or more decision variables. Furthermore, it runs almost universally faster than the other algorithms of this study, requires minimal amounts of memory, and maintains perfect median number of non-dominated solutions (NDS) throughout all trials.

Future work should focus on improving the performance of S-NSGA-II in lower decision spaces, and certain problem sub-categories (i.e., NN, SMOP4, and SMOP5). Also, none of the algorithms tested under the SMOP test suite achieved perfect HV, which is an area for improvement for all sparse LSMOPs. Also, most research in the sparse LSMOP literature has disregarded NDS as a metric for algorithms solving sparse LSMOPs. We strongly recommend that this metric become standard in the literature, as Pareto-optimal sets with identical HVs (or inverted generational distances) but different NDS are not equivalent. Also, future scholars should investigate the effects of VSSPS on other popularly-used population-based algorithms, such as particle swarm optimization and differential evolution.

By harnessing the sparsity property of LSMOPs, sparse LSMOP algorithms are able to find increasingly better solutions to this difficult class of real-world problems. In this paper, we have

adapted simple sparse operators that enable a small-scale MOP to effectively, quickly, and stably solve LSMOPs. In doing so, we are able to better train neural networks, optimize stock portfolios, and schedule crop irrigation.

7. CLIMATE-AWARE AGRICULTURAL MULTI-OBJECTIVE OPTIMIZATION USING STOCHASTIC WEATHER

7.1 Introduction

Weather variability is currently the primary source of yield variation between growing seasons (Coffel et al., 2022; Kukal and Irmak, 2018). Therefore, when agricultural inputs are scarce, data-driven resource optimization is an effective tool for improving unstable interannual yield variation (Eeswaran et al., 2021), especially evolutionary multi-objective optimization (EMO) (Kropp et al., 2019). However, though Kropp et al. (2019) provided highly precise and effective management recommendations, they could not provide farmers with predictive recommendations for the season ahead. This is because the optimization platform required accurate daily weather data for the entire growing season. Kropp et al. (2022d) demonstrated that innovization, an evolutionary multi-objective optimization technique for finding innovations (Deb and Srinivasan, 2006), can create predictive management recommendations that effectively stabilize yields within a region. However, a drawback of the methods of Kropp et al. (2022d) is that the recommendations do not provide farmers with full Pareto optimal fronts. Methodologies that output Pareto fronts are valuable, since they provide stakeholders the ability to choose the solution from the Pareto front that represents their values. That is why we aim to develop a method that uses classical EMO techniques that output a Pareto front. Therefore, in order to make these techniques work, we need a method of providing accurate daily weather predictions for the entire growing season. One such promising method is through stochastic disaggregation of monthly climate predictions (Han et al., 2017). In other words, the goal of our proposed study is to develop a platform that allows us to run classical EMO techniques for sustainable agricultural intensification using stochastically disaggregated daily weather data.

We put forth two objectives to meet this goal: *Objective 1*) generate weather that is sufficient to be used in EMO crop management models, and *Objective 2*) construct an EMO platform that prescribes a Pareto front of management actions to farmers.

7.2 Methods

7.2.1 Stochastic weather generation

The daily weather forecasts will be created through several sources, each with unique strengths. Short-term forecasts (3 days into the future) will be provided through historical daily predicted weather forecasts from NOAA's weather forecasting arm via the National Digital Forecast Database (NDFD) (NOAA, 2018). Then, sub-seasonal (i.e., "Sub-seasonal to Seasonal" (S2S)) climate predictions (2 weeks into the future) will be gathered, and then disaggregated into several daily weather realizations. This S2S forecast will come from the World Climate Research Program (Robertson et al., 2015). Finally, seasonal climate forecast (SCF) predictions will fill out the remaining growing season and come from U.S. Climate Prediction Center (CPC) (O'Lenic et al., 2008). The number of the S2S and SCF realizations is the maximum number of realizations that do not make the optimization unreasonably slow. The study area is Cass County, in Southwest Michigan. Cass County was chosen because it is a significant corn producing region that relies on irrigation (Kropp et al., 2022c). It also has daily rainfall data for the study site, as well as for several climatologically similar sites in Southwest Michigan (MSU Enviroweather, 2019).

These sources will then be compiled into rolling forecasts to generate rolling recommended management actions through the growing season. In the first rolling recommended management action, which will occur according to the physiological water demands of the cultivar, the optimization will present management actions to the farmer for the first two weeks of irrigation. Then during the second rolling management recommendations, the optimization platform will present management actions to the farmer for the 3rd through 4th weeks of irrigation actions. This process will repeat for the remainder of the growing season. During each cycle, the optimization will generate recommended management actions according to three sets of weather data. First, the short-term forecast will cover days 0 through 3. Then, the S2S prediction will be transformed into a number of stochastic realizations of days 4 through 14. Finally, the SCF forecast will be transformed into several stochastic realizations of days 15 through the remainder of the growing season. Generation of these realizations will be generated through the tools FResampler1 and predictWTD (Han et al., 2017; Han and Ines, 2017).

7.2.2 Stochastic EMO platform

As described in *Objective 1*, an optimization routine will be run every two weeks to generate recommended management practices for the growers in the study area. The required dependencies of this routine are as follows. First, a calibrated DSSAT model for maize will be chosen from the literature. In our case, we will use the calibrated DSSAT model from Jha et al. (2019; 2021) for a study site in Cass county Michigan. Next, the weather inputs will be chosen and processed as described in *Objective 1*. Finally, and most critically for this study, an objective function that can process the weather data from *Objective 1*, as well as proposed management actions, will be developed.

An objective function is an essential component of an EMO. The role of an objective function is to numerically describe the fitness of any given solution (i.e., how good of a solution it is) to a problem. In the research of Kropp et al. (2019), the objective function took a schedule of days of irrigation and the amounts of irrigation on each of those dates. The objective function would then return three objective values according to those management practices: O_f : yield, O_w :

water usage, and O_l : nitrogen leaching. However, as mentioned earlier, this function depends on historical weather data, which is not obtainable by a farmer planning a future field. Therefore, using our ensemble of possible weather realizations from *Objective 1*, we can modify this objective function to take advantage of this new data.

In our proposed method, DSSAT is run *n* times in each objective function, such that *n* is the number of weather realizations available in the ensemble (Figure 20). For each realization, the same proposed management actions are kept constant throughout the *n* DSSAT runs. Therefore, we get *n* different possible yields $(y_0, ..., y_n)$, *n* different possible leaching values $(l_0, ..., l_n)$, and one irrigation water usage (w) for the given management action (Figure 20). Ensembles of objective values are very useful since they allow us to calculate both the expected value (i.e., E(y)and E(l)) and the risk behind each management decision (i.e., $P_{fail}(y)$). Failure in this context is a threshold defined by the decision maker that is unacceptable. For example, a farmer must obtain a certain amount of yield in order to turn a profit in a given year. In summary, our proposed objective functions would be $O_{EXP(f)}$, $O_{EXP(l)}$, $O_{P_{fail}(y)}$, and O_w . Now, in the face of weather uncertainty, a farmer can see a more realistic expected value of yield and leaching, as well as the risk, behind a management decision.



Figure 20: Illustration of how an objective function can handle stochastic weather. Since the crop model can only handle one fixed season of weather, we repeatedly run the crop model under many weather realizations generated from a climate model. This, combined with the management schedule being inputted into the objective function, will result in a distribution of environmental and yield results.

7.2.3 Preliminary Run Experimental Setup

As of this dissertation, we have set up a preliminary run to demonstrate the utility of using stochastic weather in crop optimization. This preliminary run was comprised of a single (i.e., not

rolling) optimization for the entire season using a disaggregated SCF forecast. The decision variables were the amount and timing of irrigation, and the timing of a 50 kg/ha application of nitrogen. There is a constant 150 kg/ha application of nitrogen at planting. These variables were optimized to maximize average yield, minimize average leaching, minimize total irrigation applied, and minimize the risk of crop failure. The objective function calculates these values for every function evaluation by simulating the fixed management practice in DSSAT under 100 different stochastic realizations of the GCM forecast. 100 was chosen as a balance between accuracy and run time. The 100 DSSAT simulations returned an array of 100 different yields and leaching amounts, which were in turn averaged to obtain two out of the four objective values. The third objective value, total irrigation, was simply taken from the sum of irrigation applied during the season. We defined the fourth objective value, crop failure risk, as the percentage of the 100 realizations that go below a yield threshold defined by a farmer. Based on the common practices of the study area, we chose a threshold of 8,000 kg/ha as our failure threshold, but any alternative can be chosen for a different study site or decision maker. Since this is a many-objective problem, we used NSGA-III (Deb and Jain, 2014) to optimize the system. The study site was Cassopolis, in Cass County, Michigan, using the same field and DSSAT configuration as Kropp et al. (2022b).

7.3 Preliminary results

Though currently under development, this study is already yielded exciting results. We will discuss the resulting Pareto front (Figure 21) through the lens of a decision maker (i.e., a farmer) choosing a management action from the front. First, as irrigation increases, the risk of crop failure drops (Figure 21, Table 10), and crop yields rise. This makes sense since yields significantly depend on water from either rain or irrigation. Therefore, as irrigation amounts drop, more and more of the crop's water demand is satisfied solely by rain, which is highly risky. Increasing

irrigation also raises leaching levels. This agrees with the literature because irrigating at or around nitrogen applications risks leaching the nitrogen into the groundwater (Gheysari et al., 2009). In summary, irrigation management comes with delicate trade-offs. Increasing irrigation amounts can make farmers confident that their yields will be, on average, high. However, this guarantee often comes at the price of increased nitrogen leaching and water waste.



Figure 21: 4-D Pareto front of average yields, average leaching, total irrigation, and crop failure risk. The bottom center sub-figure (a) is the Pareto set head-on, the top right sub-figure (b) is the Pareto set rotated to the right, and the top left sub-figure (c) is the Pareto set rotated to the left.

Beyond being a useful illustration of the relationship between objectives, this Pareto front would be an attractive decision support tool for a farmer. To illustrate the point, imagine a hypothetical farmer looking to find a management regime for their farm amongst the Pareto optimal set (Table 10). Let's say that the farmer primarily wants to maximize their yields and profits, but also wants to minimize leaching for ethical purposes. Looking at Table 10, the obvious good choice with respects to yield is *solution 1*, since it has both the highest average yield and the lowest risk of crop failure. However, since the average leaching is quite high at 17.90 kg/ha (the minimum average leaching for the Pareto optimal set is 8.89 kg/ha), the farmer might be willing to sacrifice a bit of yield, water efficiency, and/or risk for better average leaching. A farmer following conventional management decision making would not have any way of clearly navigating this trade-off. But with the optimized Pareto front, the farmer can implement a management decision that will guarantee the desired trade-off. In this case, the farmer might choose *solution 5* (Table 10), which increases risks by 6%, decreases average yield by 273.26 kg/ha, and an increases irrigation by 2.51 mm for a 5.36 kg/ha decrease in average leaching. These changes illustrate just one way that famers can intelligently understand and choose among the possible trade-offs in a multi-objective system.

As promising as these results are, they are only the beginning of this research undertaking. The Pareto front from (Table 10) was generated using stochastic weather generated from only a seasonal forecast. Once our methodology is complete, including three types of weather forecasts and bi-weekly rolling optimization, we hypothesize that these results will improve, especially with respects of risk and environmental efficiency.

		Pareto Optin	mal Solutions	1-50		F	Pareto Optin	nal Solutions	51-100	
Sol.	Fail	Yield	Leaching Irrigation		Sol.	Fail	Yield	Leaching	Irrigation	
no.	Risk	(kg/ha)	(kg/ha)	(mm)	no.	Risk	(kg/ha)	(kg/ha)	(mm)	
1*	0.06	10,110.16	17.90	10.32	51	0.4	7,990.79	9.45	1.24	
2	0.09	9,975.09	20.41	8.32	52	0.4	7,990.79	9.45	1.24	
3	0.09	9,888.5	18.48	10.14	53	0.52	7,919.64	11.09	2.33	
4	0.12	9,863.04	12.54	15.87	54	0.57	7,907.48	14.39	2.33	
5*	0.16	9,836.9	12.54	12.83	55	0.47	7,851.34	9.26	3.22	
6	0.18	9,750.33	11.63	12.76	56	0.47	7,844.68	9.10	3.22	
7	0.13	9,733.06	14.04	8.91	57	0.57	7,835.66	12.15	3.24	
8	0.16	9,730.86	12.67	8.78	58	0.57	7,807.22	9.24	11.19	
9	0.18	9,603.69	16.55	11.75	59	0.47	7,735.01	11.36	1.18	
10	0.18	9,570.27	11.07	14.66	60	0.53	7,717.73	9.24	2.19	
11	0.18	9,567.65	12.40	9.72	61	0.6	7,700.06	14.15	5.26	
12	0.19	9,536.1	19.40	6.91	62	0.57	7,686.63	9.19	7.18	
13	0.22	9,452.55	12.67	6.65	63	0.53	7,662.59	9.41	1.18	
14	0.21	9,271.95	10.48	12.5	64	0.53	7,661.1	9.30	1.18	
15	0.25	9,256.68	12.32	3.59	65	0.53	7,661.1	9.30	1.18	
16	0.22	9,191.21	10.17	7.47	66	0.53	7,661.1	9.30	1.18	
17	0.25	9,125.47	10.26	5.46	67	0.53	7,657.03	11.16	1.16	
18	0.21	9,111.58	10.15	8.48	68	0.57	7,622.86	9.18	6.18	
19	0.33	9,074.26	16.71	4.57	69	0.61	7,574.53	14.09	13.38	
20	0.25	8,970.94	9.96	4.45	70	0.62	7,480.78	11.47	11.14	
21	0.31	8,928.04	14.28	11.47	71	0.65	7,419.5	9.67	10.18	
22	0.29	8,915.12	15.42	4.51	72	0.65	7,342.14	9.65	11.14	
23	0.27	8,909.37	11.03	3.48	73	0.57	7,342.02	9.34	4.18	
24	0.3	8,869.49	10.94	11.58	74	0.65	7,280.56	9.46	11.13	
25	0.29	8,858.8	11.60	2.49	75	0.65	7,279.75	13.03	3.29	
26	0.35	8,834.46	10.18	10.41	76	0.61	7248.65	11.44	2.12	
27	0.35	8,812.96	9.98	11.41	77	0.69	7,221.55	11.09	7.1	
28	0.31	8,792.18	10.39	6.43	78	0.65	7,165.94	11.91	5.29	
29	0.29	8,742.78	9.78	14.34	79	0.61	7,160.8	11.26	1.11	
30	0.36	8,706.07	11.12	11.38	80	0.61	7,160.8	11.26	1.11	
31	0.35	8,674.23	13.36	8.38	81	0.61	7,153.64	9.18	3.11	
32	0.36	8,655.96	11.12	10.39	82	0.62	7,140.3	9.17	8.11	
33	0.29	8,651.33	13.68	2.37	83	0.62	7,138.37	9.39	4.14	
34	0.33	8,625.11	14.25	5.41	84	0.65	7,119.94	9.46	8.11	
35	0.39	8,540.8	9.91	4.36	85	0.65	7,114.03	12.97	3.26	
36	0.35	8,363.02	9.86	8.33	86	0.61	7,084.92	9.25	3.1	
37	0.35	8,265.68	9.37	4.29	87	0.61	7078.22	9.22	1.11	
38	0.35	8,253.72	12.71	4.28	88	0.61	7,078.22	9.22	1.11	
39	0.35	8,223.31	9.34	3.29	89	0.69	7,062.35	9.05	7.1	
40	0.42	8,184.46	14.51	4.35	90	0.65	7,032.5	12.78	2.23	
41	0.48	8,166.4	12.96	6.3	91	0.65	7,029.74	13.12	1.25	
42	0.35	8,160.33	12.39	3.25	92	0.65	6,838.76	12.03	1.25	
43	0.48	8,154.03	10.05	9.28	93	0.68	6,808.8	9.19	4.02	
44	0.48	8,148.85	12.84	6.29	94	0.68	6,757.35	9.19	3.02	
45	0.35	8,095.34	9.60	2.27	95	0.69	6,739.6	10.37	1.01	
46	0.44	8,039.23	12.07	1.24	96	0.69	6,733.88	9.11	2.03	
47	0.44	8,026.63	9.40	12.22	97	0.71	6,677.95	8.89	3.03	
48	0.53	8,011.97	12.36	6.24	98	0.69	6,610.74	10.37	0	
49	0.4	7,990.79	9.45	1.24	99	0.71	6,513.46	9.19	1	
50	0.4	7,990.79	9.45	1.24	100	0.71	6,474.88	9.19	0	

Table 10: Pareto optimal solutions for climate-aware optimization run, sorted by yield.

7.4 Conclusion

Our results demonstrate the utility of incorporating stochastic weather into crop optimization, and how weather uncertainty can be integrated into multi-objective optimization through a risk objective. Furthermore, a Pareto front that includes a risk objective illustrates the trade-offs among different decisions when managing irrigation and nitrogen in crop production. Quantifying risk in this way enables farmers to predictively manage their corn for future management while also choosing the lowest risk solution for their particular circumstances. Also, unlike conventional agricultural decision making, our methods allow farmers to finely tune the balance between production and environmental objectives.

To conclude this study, several steps will need to happen. To complete *Objective 1*, we must compile the completed disaggregated weather forecasts into rolling bi-weekly forecasts. In their current state, our three separate forecasts are not combined, nor are they in a format that could support a rolling optimization routine. For *Objective 2*, we need to build the optimization framework that can provide farmers with the rolling bi-weekly Pareto fronts for management. On top of that, since the management decisions for a given two-week cycle depend on the decisions from the previous cycles, we will need to implement an automatic decision maker to choose which Pareto optimal solution is implemented every two weeks. Finally, a validation routine must compare our results to common practices in the study area, as well as other competing smart agricultural methods.

8. CONCLUSIONS

This dissertation aimed to create innovative climate-smart irrigation and nitrogen management practices through novel, cutting-edge sparse large-scale multi-objective optimization. Through this undertaking, the fields of biosystems & agricultural engineering and computer science engineering complement each other. Computer science gained new insight into sparse large-scale multi-objective problems, and biosystems engineering gained new climate-aware tools to aid farmers in sustainably meeting food demands for the future. In summary, we demonstrated:

- Seeding population-based multi-objective algorithms with sparse initial populations, which we refer to as the SPS algorithm, dramatically improves algorithm performance when solving sparse large-scale multi-objective problems.
- NSGA-II with SPS was able to find better solutions more quickly than the state-of-the-art sparse large-scale multi-objective algorithm in solving the most complex problems in this problem class.
- Innovization is an effective method of accounting for climate uncertainty in optimizing future growing seasons.
- Through innovization, we created human-centric updates to local common practices that boosted agricultural productivity with minimal environmental impact.
- We can effectively solve sparse large-scale multi-objective optimization problems by only updating the genetic operators in existing small-scale multi-objective optimization algorithms
- Our genetic operators, combined with NSGA-II, outperformed several contemporary algorithms with respect to convergence, run time, and the number of non-dominated solutions.

9. FUTURE RESEARCH RECOMMENDATIONS

Though this dissertation furthered the research fields of agriculture and biosystems engineering and computer science engineering, much work remains. We outline our future recommended research directions as follows:

- Field tests should be conducted to determine the efficacy of our innovization recommendations in Cass County, Michigan
- Currently, our research uses multi-criterion decision making (MCDM) after the optimization run. Adopting the practices of interactive EMO, where decision makers interactively perform MCDM during the optimization, is an opportunity further to speed up the rate of convergence in agricultural optimization.
- Decision tools for predictive management recommendations through EMO should be created for farmers.
- Future multi-objective optimization research should include social objectives, as well as environmental and economic objectives
- Sparse LSMOPs still have room for improvement in solving the SMOP test suit with respect to hypervolume.
- Future LSMOP research should include the number of non-dominated solutions as a metric. Many algorithms struggle to maintain full Pareto optimal sets when solving problems with many decision variables.

APPENDIX

Supplementary Materials: Benefits of Sparse Population Sampling in Multi-Objective Evolutionary Computing for Large-Scale Sparse Optimization Problems



Figure A1: Effective runs measuring the effect of varying the number of decision variables on hyper volume. Note that the variances of some configurations were so small that the confidence

intervals were rendered as lines.



Figure A2: Effective runs measuring the effect of varying the number of decision variables on NDS. Note that many solutions maintained 100 NDS under all numbers of decision variables, and overlap each other on y = 100, and that the variances of some configurations were so small

that the confidence intervals were rendered as lines.



Figure A3: Effective runs measuring the effect of varying problem sparsity on HV. Note that the variances of some configurations were so small that the confidence intervals were rendered as

lines.


Figure A4 Effective runs measuring the effect of varying problem sparsity on NDS. Note that many solutions maintained 100 NDS under all numbers of decision variables, and overlap each other on y = 100, and that the variances of some configurations were so small that the confidence

intervals were rendered as lines.



Figure A5: Comparative runs measuring the effect of varying number of decision variables on HV. Note that the variances of some configurations were so small that the confidence intervals

were rendered as lines



Figure A6: Comparative runs measuring the effect of varying number of decision variables on run time. Note that the variances of some configurations were so small that the confidence intervals were rendered as lines.



Figure A7: Comparative runs measuring the effect of varying number of decision variables on NDS. Note that many solutions maintained 100 NDS under all numbers of decision variables, and overlap each other on y = 100 and Note that the variances of some configurations were so

small that the confidence intervals were rendered as lines.



Figure A8: Example of why HV reference points of (1,1) were not used. This run measures the performance of SparseEA versus NSGA-II with SPS at varying decision variables, and uses a reference point of (1,1). After 500 decision variables, both algorithms struggle to find solutions within the box (1,1), and the HV goes to zero. Therefore, these HV values lack any useful information on how the algorithms perform in larger decision spaces beyond 500 decision

variables.

Table A1. Effective performance of NSGA-II, MOPSO, and MOEA-D-DE with varying number

		p-value of NSGA-II		p-value o	f MOPSO	p-value of MOEA-D-DE		
# dec. vars.	Test prob.	with vs wi	thout SPS	with vs without SPS		with vs wi	thout SPS	
	-	HV	NDS	HV	NDS	HV	NDS	
100	SMOP1	3.557e-21+	$_1\approx$	3.557e-21+	$_1\approx$	3.557e-21+	1.946e-01≈	
100	SMOP2	3.557e-21+	$_1\approx$	3.557e-21+	$3.254e-01 \approx$	3.557e-21+	4.260e-02 [≈]	
100	SMOP3	$1.893e-17^{+}$	$3.254e-01 \approx$	6.477e-21+	$1.511e-04^{-1}$	3.557e-21+	$4.031e-11^{-1}$	
100	SMOP4	3.557e-21+	$_1 \approx$	3.557e-21+	$_1 \approx$	3.557e - 21 +	3.913e-01≈	
100	SMOP5	3.557e-21+	3.254e-01≈	3.557e-21+	$5.471e-24^{-1}$	3.557e - 21 +	$1.186e-16^{-1}$	
100	SMOP6	$5.478e-04^{-1}$	$_1 \approx$	9.171e-21+	5.383e-24	6.803e-19+	$1.768e-14^{-1}$	
100	SMOP7	3.557e-21+	1.654e-01≈	3.557e-21+	$_1 \approx$	3.557e - 21 +	4.552e - 13 +	
100	SMOP8	3.557e-21+	2.890e-01≈	3.557e-21+	5.802e-01≈	3.557e - 21 +	$1.378e-08^{+}$	
500	SMOP1	3.557e-21+	1≈	3.557e-21+	1≈	3.557e-21 ⁺	5.669e-01≈	
500	SMOP2	3.557e-21+	$_1 \approx$	3.557e-21+	$_1 \approx$	3.557e-21+	$1.039e-03^{$	
500	SMOP3	$4.012e-20^{+}$	3.254e-01≈	3.557e-21+	$1.040e-03^{-1}$	3.557e-21+	$7.924e-23^{-1}$	
500	SMOP4	3.557e-21+	1≈	3.557e-21+	1.590e-01≈	$3.557e-21^+$	$3.646e-03^{-1}$	
500	SMOP5	3.557e-21+	$1 \approx$	3.557e-21+	5.448e-24	3.557e-21 ⁺	$1.364e-17^{-1}$	
500	SMOP6	$3.557e-21^+$	1≈	3.557e-21+	5.334e-24	2.322e-15 ⁺	$1.155e-16^{-1}$	
500	SMOP7	$3.557e-21^+$	3.254e-01≈	3.557e-21+	1≈	3.557e-21+	2.235e-03	
500	SMOP8	$3.557e-21^+$	$4.059e-11^+$	$3.557e-21^+$	8.240e-02≈	$3.557e-21^+$	2.696e-02≈	
1.000	SMOP1	3.557e-21+		3.557e-21+		3.557e-21+	5.539e-01≈	
1,000	SMOP2	3 557e-21+	1≈	$3.557e_{-21}$ +	_ 1≈	$3.557e_{-21}$ +	$1.952e-03^{-}$	
1,000	SMOP3	$2.342e-20^+$	1≈	$3.557e_{-21}+$	$1.511e-04^{-1}$	$3.557e_{-21}$ +	2.157e-23	
1,000	SMOP4	3 5570-21+	1≈	$3.557e_{-21}+$	1.0110-04	3.557e-21+	$2.343e_{-}02^{-}$	
1,000	SMOP4	3.5570-21+	1≈	3 5570-21+	5 3490-34	$3.557e^{-21}$	2.343e-02	
1,000	SMOPS	3.5570-21+	1≈	3.5570-21+	5 4200-24	$3.557e^{-21}$	5.0520-14	
1,000	SMOP7	3.5576-21	1≈	3.5576-21	1≈	3.5576-21	7.007.08	
1,000	SMOP8	3 557e-21+	5 503e-04+	$3.557e_{-21} +$	$3254e-01 \approx$	$3.557e_{-21}$ +	4 205e-04	
2,500	SMOP1	3 5570-21+	1≈	3 5570-21+	1≈	3 5570-21+	6.7860-03	
2,500	SMOP1	3 5570-21+	1≈	3 5570-21+	1 1≈	3 5570-21+	4.015e-05	
2,500	SMOP2	3 5570-21+	1≈	3 5570-21+	4.0180-05-	3 5570-21+	5.6440-24	
2,500	SMOP3	3 5570-21+	1≈	3 5570-21+	4.018e-03	3 5570 21+	$3.3430.02^{-1}$	
2,500	SMOP4	$3.557e-21^+$	1 1≈	3.557e-21	1.390e-01	3.557e-21	2.343e-02	
2,500	SMOP5	$3.557e^{-21}$	1≈	$3.557e^{-21}$	5 2050 24	$1.728 \cdot 18^+$	3.7376-14	
2,500	SMOP7	3.557e-21	1 1≈	$3.557e^{-21}$	3.203€-24	4 245 - 21+	2.8910-10	
2,500	SMOP	$3.557e-21^+$	1 1≈	3.557e-21	1 1≈	4.3436-21	2.439e-09	
2,300	SMOP	3.557e-21 +	1≈	3.557e-21 +	1	3.557e-21	2.0986-12	
5,000	SMOPI	3.557e-21	1~	3.557e-21	1~	3.557e-21	2.342e-02	
5,000	SMOP2	3.557e-21	1~	3.557e-21	1 001 007	3.557e-21	1.248e-06	
5,000	SMOP3	3.557e-21	1 ~	2.123e-20 ⁺	1.261€-02	3.557e-21	2.1596-23	
5,000	SMOP4	3.557e-21	1~	3.557e-21	T 100- 01-	3.557e-21	1.261e-02	
5,000	SMOP5	3.557e-21	1~	3.557e-21	5.482e-24	3.557e-21	6.064e-14	
5,000	SMOPE	3.557e-21	1~	3.557e-21	5.422e-24 .≈	7.921e-20	7.985e-15	
5,000	SMOP7	3.557e-21	1~	3.557e-21	1.0	6.216e-20	3.762e-12	
5,000	SMOP8	3.557e-21+	1~	3.557e-21+	4.370e-02~	3.557e-21	2.953e-16	
7,500	SMOP1	3.557e-21+	1~	3.557e-21+	1~ 	3.557e-21	$3.254e-01^{\sim}$	
7,500	SMOP2	3.557e-21 ⁺	1~	3.557e-21 ⁺	3.254e-01∼	3.557e-21 ⁺	7.822e-05	
7,500	SMOP3	$3.557e-21^{-1}$	1.590e-01~ .~	$3.557e-21^{-1}$	$4.018e-05^{}$	3.557e-21 ⁺⁺	8.028e-23	
7,500	SMOP4	3.557e-21 ⁺	1~ .~	3.557e-21 ⁺	3.254e-01∼	3.557e-21 ⁺⁺	6.783e-03	
7,500	SMOP5	$3.557e-21^+$	1~	$3.557e-21^+$	5.397e-24	$3.557e-21^+$	1.026e-15	
7,500	SMOP6	3.557e-21 ⁺	1~ 1~	3.557e-21 ⁺	5.402e-24	2.094e-18 ⁺⁺	5.783e-14	
7,500	SMOP7	3.557e-21 ⁺	1~	3.557e-21 ⁺	1~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	6.849e-20 ⁺	$1.806e-09^{-1}$	
7,500	SMOP8	$3.557e-21^+$	$_{1}\sim$	$3.557e-21^+$	$1.590e-01 \approx$	$3.557e-21^+$	$1.540e-16^{}$	

of decision variables.

Let $-, \approx$, and + denote statistical differences between the runs with and without SPS according to the Wilcoxon Rank-Sum test. - denotes configurations that performed worse with SPS, + denotes configurations that performed better with SPS, and \approx marks no difference between the algorithm with or without SPS.

				-
# dec. vars.	Test prob.		p-values	NDG
100		HV	Run time	NDS
100	SMOP1	5.601e-10 ⁺	1.379e-13	1~ ≈
100	SMOP2	7.350e-01~	3.497e-21+	4.369e-02∼
100	SMOP3	8.179e-05	3.547e-21+	3.254e-01∼
100	SMOP4	$6.945 e{-17}$	6.050e-16+	$1 \sim$
100	SMOP5	3.931e-21	2.762e-18+	$1 \sim \sim$
100	SMOP6	3.557e-21	$4.485e-18^{+}$	1~
100	SMOP7	$1.739e-05^+$	$2.420e-15^+$	$1.260e-02^+$
100	SMOP8	$7.546e-20^{-1}$	3.539e-21+	5.570e-16 ⁺
500	SMOP1	6.849e-20	3.547e-21+	$_{1}\approx$
500	SMOP2	$3.557e-21^{-1}$	3.519e-21+	$_1\approx$
500	SMOP3	$3.557e-21^{}$	3.526e-21+	$_1\approx$
500	SMOP4	$1.102 e-07^{-}$	$3.544e-21^+$	$_1\approx$
500	SMOP5	$6.477 \mathrm{e}{-21}^{-1}$	3.550e-21+	$_1\approx$
500	SMOP6	$3.557 \text{e-} 21^{-1}$	3.551e-21+	$_1\approx$
500	SMOP7	$3.557 \text{e-} 21^{-1}$	3.550e-21+	$3.254\mathrm{e}{-01} \approx$
500	SMOP8	$3.557 \text{e-} 21^{-}$	3.535e-21+	$1.213e-09^+$
1,000	SMOP1	$7.350e-01 \approx$	3.550e-21+	$_1\approx$
1,000	SMOP2	$4.875e-20^{-1}$	3.538e-21 +	$_1\approx$
1,000	SMOP3	$2.282e-19^{-1}$	3.543e-21 +	$_1\approx$
1,000	SMOP4	7.315e-04	3.546e-21 +	$_1\approx$
1,000	SMOP5	3.998e-04 +	3.551e-21+	$_1\approx$
1,000	SMOP6	$3.557e-21^{-1}$	3.552e-21+	$_1\approx$
1,000	SMOP7	$1.744e-20^{-1}$	3.547e-21 +	$_1\approx$
1,000	SMOP8	$3.557e-21^{-1}$	3.536e-21+	$_{3.254e-01} \approx$
2,500	SMOP1	1.149e-17 ⁺	3.534e-21+	1≈
2,500	SMOP2	9.988e-02≈	3.543e-21 +	$_1\approx$
2,500	SMOP3	3.392e-15	3.545e-21 +	$_1 \approx$
2,500	SMOP4	5.514e-01≈	3.531e-21+	$_1 \approx$
2,500	SMOP5	$9.472e-17^+$	3.548e-21 +	$_1\approx$
2,500	SMOP6	$3.557e-21^{-1}$	3.553e-21 +	$_1\approx$
2,500	SMOP7	1.229e-15+	3.555e-21+	$_1\approx$
2,500	SMOP8	$1.660e-20^{-1}$	3.549e-21+	$_1 \approx$
5,000	SMOP1	3.557e-21+	3.548e-21+	3.622e-24+
5,000	SMOP2	3.557e-21+	3.553e-21+	4.016e-24 ⁺
5,000	SMOP3	6.896e-02≈	3.552e-21+	4.987e-24+
5,000	SMOP4	3.557e-21+	5.496e-01≈	4.549e-24 ⁺
5,000	SMOP5	3.557e-21+	3.554e-21+	4.4826-24+
5,000	SMOP6	3 557e-21 ⁺	3 552e-21+	5 131e-24 ⁺
5,000	SMOP7	3.557e-21 ⁺	3.552e-21+	$3.024e-24^+$
5,000	SMOP8	3.557e-21+	3.553e-21+	$3.721e-24^+$
7,500	SMOP1	3 5570-21+	3 555-21+	4 0390-24+
7,500	SMOP2	3 557e-21+	3 554e-21+	4 009e-24+
7,500	SMOP2	4.025 01≈	3 5540 21+	1 0230 224
7,500	SMOP4	3 5570 21+	3 5540 21+	4 4780 24+
7,500	SMOP4	3 5570 21+	3 5540-21+	4.3800-24+
7,500	SMOP	3.5570-21+	2 5550 21+	4.8050-24+
7,500	SMOP7	3 5570-21+	3 5530-21+	3.0190-24+
7,500	SMOP	4 5680-21+	3.5540-21+	3.0460-24+
7,000	SMOPS	*1.000e=21	0.0046-21	0.9406-24

Table A2. p-value of two-sided Wilcoxon Rank-Sum test for the comparative runs.

Let $-, \approx$, and + denote statistical differences between SparseEA and NSGA-II with SPS according to the Wilcoxon Rank-Sum test. - denotes that SparseEA performed better, + denotes NSGA-II with SPS performed better, and \approx marks no statistical difference between SparseEA and NSGA-II with SPS

Supplementary Materials: Agricultural Innovization: An Optimization-Driven Solution for Sustainable Agricultural Intensification in Michigan

S1 Cultivar coefficients

A critical first step to effectively employing the crop model is to accurately calculate the cultivar coefficients for CERES-Maize. Originally developed by Jones and Kiniry (1986), the current version of the CERES-Maize model evaluates maize management practices and production and is included in the DSSAT suite of models (Li et al., 2015; Liu et al., 2011; Lizaso et al., 2011). As with all models, the parameters of CERES-Maize must be calibrated and validated. A significant set of parameters that drive the model performance are the crop-specific parameters widely known as genetic coefficients. Therefore, calibration of the genetic coefficients is required with every new cultivar introduced to the model (Jones et al., 2011). We selected cultivar coefficients of popular maize hybrids for Cassopolis, which were validated for the study area. (Jha, 2019; Jha et al., 2021) estimated these coefficients using an ensemble of three coefficient estimation methods: a.) GENotype Coefficient cALCculator (GENCALC) (Bao et al., 2017; Hunt et al., 1993); b.) Generalized Likelihood Uncertainty Estimation (GLUE) (He et al., 2009; Sun et al., 2016) and c.) Noisy Monte Carlo Genetic Algorithm (NMCGA) (Ines & Mohanty, 2008; Jha et al., 2021).

GENCALC calculates the genetic coefficients of new cultivars using a gradient search technique using crop, weather, and soil data input for the genotype. However, it does not account for parameter uncertainty. GLUE overcomes this limitation by using observed data and a priori information about parameter distributions to estimate posterior distributions of model parameters. The prior and posterior distributions are computed based on Bayes' theorem (Bolstad & Curran, 2016), which is used to understand uncertainties in parameters and model outputs. Another method is Noisy Monte Carlo Genetic Algorithm (NMCGA) (Ines & Mohanty, 2008) which employs a Monte Carlo approach to evaluate realizations of model parameters with assumed distributions. The process of basic genetics (selection, crossover, and mutation) is employed at each solution of model parameter for every generation until a solution is achieved, which is judged based on the goodness-of-fit of the model. The coefficients chosen for this study are an ensemble (weighted average) of all three methods, which yielded the best performing parameters and can be adopted across the geographical location for the validation studies. The weights were assigned based on inverse squared distance between the predicted and observed variable(s) of phenology and growth, which is substantially impacted by respective phenology or growth coefficients (Jha et al., 2021).



Figure A9. Complete map of all validation weather stations



Figure A10. Example of Pareto fronts from 1991 and 1994

Table A3. Growing degree day in Fahrenheit during the innovization period for the vegetative

Innovization	Vegetative and Reproductive Development Stages (number of days after planting)														
Years*	V6	V7	V8	V9	V10	V11	V12	V13	V14	R1	R2	R3	R4	R5	R6
1989 ⁿ	61	65	69	72	78	80	83	85	87	100	110	120	126	132	162
1990 ^w	57	61	65	71	75	78	81	84	86	99	109	119	125	131	161
1991^{d}	46	50	54	58	62	65	68	71	74	84	94	104	110	116	146
1992 ⁿ	62	66	70	75	80	84	87	90	93	108	118	128	134	140	170
1993 ^w	60	64	68	72	77	80	83	86	89	100	110	120	126	132	162
1994 ⁿ	55	59	63	66	71	74	77	80	83	94	104	114	120	126	156
1995 ⁿ	61	65	69	72	78	81	84	87	90	102	112	122	128	134	164
1996 ^w	58	62	66	70	76	78	81	84	87	100	110	120	126	132	162
1997 ^w	66	70	74	77	84	87	90	93	96	107	117	127	133	139	169
1998 ^d	53	57	61	65	69	71	74	77	80	91	101	111	117	123	153
1999 ^d	50	54	58	64	70	72	75	78	81	91	101	111	117	123	153
2000^{n}	53	57	61	66	71	74	77	80	83	96	106	116	122	128	158
2001 ^w	54	58	62	66	73	75	78	81	84	97	107	117	123	129	159
2002^{d}	59	63	67	71	75	77	80	83	86	99	109	119	125	131	161
2003^{d}	61	65	69	73	78	80	83	86	89	103	113	123	129	135	165
2004^{n}	47	51	55	60	65	69	72	75	78	91	101	111	117	123	153
2005^{d}	52	56	60	67	71	73	76	79	82	96	106	116	122	128	158
2006 ^w	55	59	63	67	71	74	77	80	83	97	107	117	123	129	159
2007 ⁿ	50	54	58	63	66	69	72	75	78	90	100	110	116	122	152
2008 ^w	53	57	61	69	74	76	79	82	85	100	110	120	126	132	162
2009 ⁿ	58	62	66	70	74	76	79	82	85	101	111	121	127	133	163
2010^{n}	52	56	60	65	69	71	74	77	80	94	104	114	120	126	156
2011 ^w	57	61	65	70	75	78	81	84	87	106	116	126	132	138	168
2012^{d}	53	57	61	66	70	73	76	79	82	96	106	116	122	128	158
2013 ⁿ	57	61	65	70	74	76	79	82	85	100	110	120	126	132	162
2014^{n}	56	60	64	68	73	75	78	81	84	98	108	118	124	130	160
2015 ⁿ	54	58	62	67	72	74	77	81	84	97	107	117	123	129	159
2016 ⁿ	55	59	63	67	72	74	77	81	84	99	109	119	125	131	161
2017 ^w	55	59	63	67	73	77	80	83	86	102	112	122	128	134	164
2018 ^w	60	64	60	64	69	71	74	77	80	93	103	113	119	125	155

and reproductive development stages

* A superscript *d* denotes a dry year of precipitation, a superscript *n* denotes a normal year of precipitation, and a superscript *w* denotes a wet year of precipitation.

Table A4. The *p*-values of the innovization strategies over the *common practice strategy* for all validation seasons as well as over different climate scenarios. Superscripts +, -, and \approx denote a statistical improvement, statistical worsening, or no statistical change from the *common practice*

	Yield	Leaching	Water Efficiency		
Strategy	p-values	p-values	p-values		
	Per	formance acro	oss all years		
Irrigation recommendation	3.51%+	64.45% [≈]	$0.31\%^-$		
Nitrogen recommendation	$90.46\%^{pprox}$	95.89% [≈]	$90.90\%^{pprox}$		
All recommendations	3.21%+	61.13% [≈]	$0.30\%^{-}$		
	Perf	ormance duri	ng dry years		
Irrigation recommendation	38.86% [≈]	$89.14\%^{pprox}$	15.32% [≈]		
Nitrogen recommendation	99.91% [≈]	93.26% [≈]	$89.71\%^{pprox}$		
All recommendations	38.83% [≈]	$80.99\%^{pprox}$	$16.45\%^{pprox}$		
	Perfor	mance during	g normal years		
Irrigation recommendation	28.25% [≈]	88.37% [≈]	$0.09\%^-$		
Nitrogen recommendation	$96.82\%^{pprox}$	$87.26\%^{pprox}$	$100.00\%^{pprox}$		
All recommendations	$24.41\%^{pprox}$	91.53% [≈]	$0.09\%^-$		
	Perf	ormance duri	ng wet years		
Irrigation recommendation	2.83%+	48.25% [≈]	5.11% ⁻		
Nitrogen recommendation	90.64% [≈]	94.92% [≈]	$99.67\%^{pprox}$		
All recommendations	$2.94\%^{+}$	$49.07\%^{pprox}$	5.01% ⁻		

strategy, respectively.

Supplementary Materials: Improved Evolutionary Operator for Sparse Large-Scale Multi-Objective Optimization Problems



Figure A11. Distribution of nonzero values throughout a population of 100 with 1,000 decision variables from SPSS.



Figure A12. Target sparsities versus actual sparsities realized in SSPS. As θ increases, the gap calculation increasingly affects the accuracy of the realized sparsity in the population.



Figure A13. The distribution of sparsity after repeatedly performing S-SBX on the same initial population 1,000 times. The initial population is generated through SSPS with a target sparsity of 0.25. See how the distribution of resulting sparsities are distributed around the initial sparsity, similar to the distribution for SBX or PM.



Figure A14: Performance (i.e., mean HV) of the studied algorithms solving SMOP1 with increasing numbers of decision variables. The shaded areas around each mean line is the 95% confidence interval for the mean.



Figure A15: Performance (i.e., mean HV) of the studied algorithms solving SMOP2 with increasing numbers of decision variables. The shaded areas around each mean line is the 95% confidence interval for the mean.



Figure A16: Performance (i.e., mean HV) of the studied algorithms solving SMOP3 with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.



Figure A17: Performance (i.e., mean HV) of the studied algorithms solving SMOP4 with increasing numbers of decision variables. The shaded areas around each mean line are the 95%

confidence interval for the mean.



Figure A18: Performance (i.e., mean HV) of the studied algorithms solving SMOP5 with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.



Figure A19: Performance (i.e., mean HV) of the studied algorithms solving SMOP6 with increasing numbers of decision variables. The shaded areas around each mean line are the 95%

confidence interval for the mean.



Figure A20: Performance (i.e., mean HV) of the studied algorithms solving SMOP7 with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.



Figure A21: Performance (i.e., mean HV) of the studied algorithms solving SMOP8 with increasing numbers of decision variables. The shaded areas around each mean line are the 95%

confidence interval for the mean.



Figure A22: Performance (i.e., mean HV) of the studied algorithms solving the NN training problem, data set 1, with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.



Figure A23: Performance (i.e., mean HV) of the studied algorithms solving the NN training problem, data set 2, with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.



Figure A24: Performance (i.e., mean HV) of the studied algorithms solving the NN training problem, data set 3, with increasing numbers of decision variables. The shaded areas around each mean line are the 95% confidence interval for the mean.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Abbasi, A.Z., Islam, N., Shaikh, Z.A., others, 2014. A review of wireless sensors and networks' applications in agriculture. Computer Standards & Interfaces 36, 263–270.
- Abendroth, L.J., Elmore, R.W., Boyer, M.J., Marlay, S.K., 2010. Understanding corn development: A key for successful crop management.
- Ahmed, N., De, D., Hussain, I., 2018. Internet of Things (IoT) for smart precision agriculture and farming in rural areas. IEEE Internet of Things Journal 5, 4890–4899.
- Akbari, M., Gheysari, M., Mostafazadeh-Fard, B., Shayannejad, M., 2018. Surface irrigation simulation-optimization model based on meta-heuristic algorithms. Agricultural Water Management 201, 46–57.
- Alo, A.O., Baines, R., Conway, J., Cannon, N., 2017. The Impacts of Climate Change on Agriculture in Developing Countries: A Case Study of Oyo State, Nigeria. The International Journal of Climate Change: Impacts and Responses 9, 1–21.
- Andresen, J., Olsen, L., Aichele, T., Bishop, B., Brown, J., Landis, J., Marquie, S., Pollyea, A., 2011. Enviro-weather: A weather-based pest and crop management information system for Michigan, in: Proc. 7th International Integrated Pest Management Symposium, Memphis, TN. pp. 27–29.
- Antonio, L.M., Coello Coello, C.A., 2013. Use of cooperative coevolution for solving large scale multiobjective optimization problems. IEEE, pp. 2758–2765.
- Araya, A., Prasad, P., Ciampitti, I., Jha, P., 2021. Using crop simulation model to evaluate influence of water management practices and multiple cropping systems on crop yields: A case study for Ethiopian highlands. Field Crops Research 260, 108004.
- Arias, P., Bellouin, N., Coppola, E., Jones, R., Krinner, G., Marotzke, J., Naik, V., Palmer, M., Plattner, G.-K., Rogelj, J., others, 2021. Climate Change 2021: The Physical Science Basis. Contribution of Working Group14 I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change; Technical Summary.
- Awange, J., 2022. Global freshwater resources, in: Food Insecurity & Hydroclimate in Greater Horn of Africa. Springer, pp. 67–83.
- Ban, H.-Y., Ahn, J.-B., Lee, B.-W., 2019. Assimilating MODIS data-derived minimum input data set and water stress factors into CERES-Maize model improves regional corn yield predictions. PloS one 14, e0211874–e0211874.
- Basso, B., 2020. Methods and systems for precision crop management. US20200065911A1.

- Branke, J., Scheckenbach, B., Stein, M., Deb, K., Schmeck, H., 2009. Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. European Journal of Operational Research 199, 684–693. https://doi.org/10.1016/j.ejor.2008.01.054
- Cao, B., Zhao, J., Lv, Z., Liu, X., 2017. A Distributed Parallel Cooperative Coevolutionary Multiobjective Evolutionary Algorithm for Large-Scale Optimization. IEEE transactions on industrial informatics 13, 2030–2038.
- Carolan, M., 2016. The sociology of food and agriculture. Routledge.
- Censor, Y., 1977. Pareto optimality in multiobjective problems. Applied mathematics & optimization 4, 41–59.
- Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B., 2017. Test Problems for Large-Scale Multiobjective and Many-Objective Optimization. IEEE transactions on cybernetics 47, 4108–4121.
- Chvatal, Vasek, Chvatal, Vaclav, others, 1983. Linear programming. Macmillan.
- Coello, C.A.C., 2006. Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine 1, 28–36. https://doi.org/10.1109/MCI.2006.1597059
- Coello Coello, C.A., González Brambila, S., Figueroa Gamboa, J., Castillo Tapia, M.G., Hernández Gómez, R., 2020. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. Complex & Intelligent Systems 6, 221–236. https://doi.org/10.1007/s40747-019-0113-4
- Coffel, E.D., Lesk, C., Winter, J.M., Osterberg, E.C., Mankin, J.S., 2022. Crop-climate feedbacks boost US maize and soy yields. Environmental Research Letters.
- Curry, H.B., 1944. The method of steepest descent for non-linear minimization problems. Quarterly of Applied Mathematics 2, 258–261.
- da Fonseca, E.P.R., Caldeira, E., Ramos Filho, H.S., e Oliveira, L.B., Pereira, A.C.M., Vilela, P.S., 2020. Agro 4.0: A data science-based information system for sustainable agroecosystem management. Simulation Modelling Practice and Theory 102, 102068.
- de Abreu Resenes, J., Pavan, W., Hölbig, C.A., Fernandes, J.M.C., Shelia, V., Porter, C., Hoogenboom, G., 2019. jDSSAT: A JavaScript Module for DSSAT-CSM integration. SoftwareX 10, 100271.
- Deb, K., Agrawal, R.B., 1995. Simulated Binary Crossover for Continuous Search Space. Complex Systems 9, 115–148. https://doi.org/10.1.1.26.8485Cached
- Deb, K., Jain, H., 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Transactions on Evolutionary Computation 18, 577–601. https://doi.org/10.1109/TEVC.2013.2281535

- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197. https://doi.org/10.1109/4235.996017
- Deb, K., Srinivasan, A., 2006. Innovization: Innovating design principles through optimization, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. pp. 1629–1636.
- Deb, Kalyanmoy., 2009. Multi-objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester, UK.
- D'Errico, G., Cerri, T., Pertusi, G., 2011. Multi-objective optimization of internal combustion engine by means of 1D fluid-dynamic models. Applied Energy 88, 767–777. https://doi.org/10.1016/j.apenergy.2010.09.001
- Deshcherevskaya, O., Avilov, V., Dinh, B.D., Tran, C.H., Kurbatova, J., 2013. Modern climate of the Cát Tiên National Park (Southern Vietnam): climatological data for ecological studies. Izvestiya, Atmospheric and Oceanic Physics 49, 819–838.
- Doran, J.W., 2002. Soil health and global sustainability: translating science into practice. Agriculture, ecosystems & environment 88, 119–127.
- Dua, D., Graff, C., 2017. UCI Machine Learning Repository.
- Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science 39–43. https://doi.org/10.1109/MHS.1995.494215
- Eeswaran, R., Nejadhashemi, A.P., Kpodo, J., Curtis, Z.K., Adhikari, U., Liao, H., Li, S.-G., Hernandez-Suarez, J.S., Alves, F.C., Raschke, A., others, 2021. Quantification of resilience metrics as affected by conservation agriculture at a watershed scale. Agriculture, Ecosystems & Environment 320, 107612.
- Eftekharian, S., Shojafar, M., Shamshirband, S., 2017. 2-Phase NSGA II: An Optimized Reward and Risk Measurements Algorithm in Portfolio Optimization. Algorithms 10, 130.
- El Bilali, H., Callenius, C., Strassner, C., Probst, L., 2019. Food and nutrition security and sustainability transitions in food systems. Food and energy security 8, e00154-n/a.
- Essick, P., Charles, D., 2013. Our Fertilized World. National Geographic Magazine.
- Fischer, R., Byerlee, D., Edmeades, G., 2014. Crop yields and global food security. ACIAR: Canberra, ACT 8–11.
- Gheysari, M., Mirlatifi, S.M., Homaee, M., Asadi, M.E., Hoogenboom, G., 2009. Nitrate leaching in a silage maize field under different irrigation and nitrogen fertilizer rates. Agricultural water management 96, 946–954.

- Goldberg, D.E., 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston.
- Gong, M., Liu, J., Li, H., Cai, Q., Su, L., 2015. A multiobjective sparse feature learning model for deep neural networks. IEEE transactions on neural networks and learning systems 26, 3263–3277.
- Gramig, B.M., Massey, R., Do Yun, S., 2017. Nitrogen application decision-making under climate risk in the U.S. Corn Belt. Climate risk management 15, 82–89.
- Groot, J.C.J., Oomen, G.J.M., Rossing, W.A.H., 2012. Multi-objective optimization and design of farming systems. Agricultural Systems 110, 63–77. https://doi.org/10.1016/j.agsy.2012.03.012
- Han, E., Ines, A.V., 2017. Downscaling probabilistic seasonal climate forecasts for decision support in agriculture: A comparison of parametric and non-parametric approach. Climate Risk Management 18, 51–65.
- Han, E., Ines, A.V., Baethgen, W.E., 2017. Climate-Agriculture-Modeling and Decision Tool (CAMDT): A software framework for climate risk management in agriculture. Environmental modelling & software 95, 102–114.
- Hansen, J., Challinor, A., Ines, A., Wheeler, T., Moron, V., 2006. Translating climate forecasts into agricultural terms: advances and challenges. Climate Research 33, 27–41. https://doi.org/10.3354/cr033027
- He, C., Li, L., Tian, Y., Zhang, X., Cheng, R., Jin, Y., Yao, X., 2019. Accelerating Large-Scale Multiobjective Optimization via Problem Reformulation. IEEE transactions on evolutionary computation 23, 949–961.
- Herman, M.R., Hernandez-Suarez, J.S., Nejadhashemi, A.P., Kropp, I., Sadeghi, A.M., 2020. Evaluation of multi-and many-objective optimization techniques to improve the performance of a hydrologic model using evapotranspiration remote-sensing data. Journal of Hydrologic Engineering 25, 04020006.
- Hernandez-Suarez, J.S., Nejadhashemi, A.P., Kropp, I.M., Abouali, M., Zhang, Z., Deb, K., 2018. Evaluation of the impacts of hydrologic model calibration methods on predictability of ecologically-relevant hydrologic indices. Journal of hydrology 564, 758–772.
- Herrera, F., Lozano, M., Verdegay, J.L., 1998. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial intelligence review 12, 265–319.
- Hollander, M., Wolfe, D.A., Chicken, E., 2013. Nonparametric statistical methods. John Wiley & Sons.
- Hong, W.-J., Yang, P., Tang, K., 2021. Evolutionary computation for large-scale multi-objective optimization: A decade of progresses. International Journal of Automation and Computing 18, 155–169.

- Hoogenboom, G., Porter, C.H., Shelia, V., Boote, K.J., Singh, U., White, J.W., Hunt, L.A., Ogoshi, R., Lizaso, J.I., Koo, J., 2017. Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.7.
- Ines, A.V., Mohanty, B.P., 2008. Parameter conditioning with a noisy Monte Carlo genetic algorithm for estimating effective soil hydraulic properties from space. Water resources research 44.
- Jha, P.K., 2019. Agronomic management of corn using seasonal climate predictions, remote sensing and crop simulation models.
- Jha, P.K., Ines, A.V.M., Singh, M.P., 2021. A multiple and ensembling approach for calibration and evaluation of genetic coefficients of CERES-Maize to simulate maize phenology and yield in Michigan. Environmental modelling & software : with environment data news 135.
- Jha, P.K., Kumar, S.N., Ines, A.V.M., 2018. Responses of soybean to water stress and supplemental irrigation in upper Indo-Gangetic plain: Field experiment and modeling approach. Field crops research 219, 76–86.
- Jin, Z., Archontoulis, S.V., Lobell, D.B., 2019. How much will precision nitrogen management pay off? An evaluation based on simulating thousands of corn fields over the US Corn-Belt. Field Crops Research 240, 12–22.
- Johr, H., 2012. Where are the future farmers to grow our food? International Food and Agribusiness Management Review 15, 9–11.
- Jones, Antle, J.M., Basso, B., Boote, K.J., Conant, R.T., Foster, I., Godfray, H.C.J., Herrero, M., Howitt, R.E., Janssen, S., others, 2017. Toward a new generation of agricultural system data, models, and knowledge products: State of agricultural systems science. Agricultural systems 155, 269–288.
- Jones, J., Hoogenboom, G., Porter, C., Boote, K., Batchelor, W., Hunt, L., Wilkens, P., Singh, U., Gijsman, A., Ritchie, J., 2003. The DSSAT cropping system model. European Journal of Agronomy 18, 235–265. https://doi.org/10.1016/S1161-0301(02)00107-7
- Kelley, L., 2016. Peak water use needs for corn. MSU Extension.
- Kelley, L., Hall, B., 2020. July/August corn water needs [WWW Document]. Corn. URL https://www.canr.msu.edu/news/july-august-corn-water-needs (accessed 11.12.21).
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks. IEEE, pp. 1942–1948.
- Khan, S., Tariq, R., Yuanlai, C., Blackwell, J., 2006. Can irrigation be sustainable? Agricultural water management 80, 87–99.
- Kharas, H., Gertz, G., 2010. The new global middle class: A cross-over from West to East. Wolfensohn Center for Development at Brookings 1–14.

- Kranz, W.L., Irmak, S., van Donk, S.J., Yonts, C.D., Martin, D.L., 2008. Irrigation Management for Corn [WWW Document]. URL https://extensionpublications.unl.edu/assets/html/g1850/build/g1850.htm (accessed 11.12.21).
- Kropp, I., Nejadhashemi, A.P., Deb, K., 2022a. Benefits of sparse population sampling in multiobjective evolutionary computing for large-Scale sparse optimization problems. Swarm and Evolutionary Computation 69, 101025. https://doi.org/10.1016/j.swevo.2021.101025
- Kropp, I., Nejadhashemi, A.P., Deb, K., Abouali, M., Roy, P.C., Adhikari, U., Hoogenboom, G., 2019. A multi-objective approach to water and nutrient efficiency for sustainable agricultural intensification. Agricultural Systems 173, 289–302. https://doi.org/10.1016/j.agsy.2019.03.014
- Kropp, I., Nejadhashemi, A.P., Jha, P., Hernandez-Suarez, J.S., 2022b. Agricultural Innovization: An Optimization-Driven solution for sustainable agricultural intensification in Michigan. Computers and Electronics in Agriculture 199, 107143. https://doi.org/10.1016/j.compag.2022.107143
- Kropp, I., Nejadhashemi, A.P., Jha, P., Hernandez-Suarez, J.S., 2022c. Agricultural Innovization: An Optimization-Driven Solution for Sustainable Agricultural Intensification (under review). Computers and Electronics in Agriculture.
- Kropp, I., Nejadhashemi, P., Jha, P., 2022d. Agricultural Innovization: An Optimization-Driven Solution for Sustainable Agricultural Intensification (under review). Computers and Electronics in Agriculture Under review.
- Kucharik, C.J., Ramiadantsoa, T., Zhang, J., Ives, A.R., 2020. Spatiotemporal trends in crop yields, yield variability, and yield gaps across the USA. Crop Science 60, 2085–2101.
- Kukal, M.S., Irmak, S., 2018. Climate-driven crop yield and yield variability and climate change impacts on the US Great Plains agricultural production. Scientific reports 8, 1–18.
- Li, Z.T., Yang, J.Y., Drury, C.F., Hoogenboom, G., 2015. Evaluation of the DSSAT-CSM for simulating yield and soil organic C and N of a long-term maize and wheat rotation experiment in the Loess Plateau of Northwestern China. Agricultural Systems 135, 90– 104. https://doi.org/10.1016/j.agsy.2014.12.006
- Lin, Q., Li, J., Du, Z., Chen, J., Ming, Z., 2015. A novel multi-objective particle swarm optimization with multiple search strategies. European Journal of Operational Research 247, 732–744.
- Lobao, L., Stofferahn, C.W., 2008. The community effects of industrialized farming: Social science research and challenges to corporate farming laws. Agriculture and Human Values 25, 219–240.

Lowrance, R., Stinner, B.R., House, G.J., 1984. Agricultural ecosystems: unifying concepts.

- Ma, L., Huang, M., Yang, S., Wang, R., Wang, X., 2021. An Adaptive Localized Decision Variable Analysis Approach to Large-Scale Multiobjective and Many-Objective Optimization. IEEE transactions on cybernetics PP, 1–13.
- Ma, X., Liu, F., Qi, Y., Wang, X., Li, L., Jiao, L., Yin, M., Gong, M., 2015. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. IEEE Transactions on Evolutionary Computation 20, 275–298.
- Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization 26, 369–395. https://doi.org/10.1007/s00158-003-0368-6
- Martinez-Feria, R.A., Basso, B., Great Lakes Bioenergy Research Center (GLBRC), W. (United S., Madison, 2020. Unstable crop yields reveal opportunities for site-specific adaptations to climate variability. Scientific reports 10, 2885.
- MATLAB, 2022. version 9.12.0 (R2022a). The MathWorks Inc., Natick, Massachusetts.
- MATLAB, 2020. version 9.9.0 (R2020b). The MathWorks Inc., Natick, Massachusetts.
- Mauseth, J.D., 2014. Botany: an introduction to plant biology. Jones & Bartlett Publishers.
- Mello Jr, A.V., Schardong, A., Pedrotti, A., 2013. Multi-Objective Analysis Applied to an Irrigated Agricultural System on Oxisols. Transactions of the ASABE 56, 71–82.
- Menne, M.J., Durre, I., Korzeniewski, B., McNeal, S., Thomas, K., Yin, X., Anthony, S., Ray, R., Vose, R.S., Gleason, B.E., others, 2012. Global historical climatology network-daily (GHCN-Daily), Version 3. NOAA National Climatic Data Center 10, V5D21VHZ.

Miettinen, K., 2012. Nonlinear multiobjective optimization. Springer Science & Business Media.

- Monzon, J.P., Calviño, P., Sadras, V.O., Zubiaurre, J., Andrade, F.H., 2018. Precision agriculture based on crop physiological principles improves whole-farm yield and profit: A case study. European journal of agronomy 99, 62–71.
- MSU Enviroweather, 2019. Michigan State University Enviroweather Program.
- NOAA, 2018. National Digital Forecast Database [WWW Document]. URL https://nomads.ncdc.noaa.gov/data/ndfd/
- NRCS, 2019. Soil survey staff, natural resources conservation service, United States department of agriculture. Soil Survey Geographic (SSURGO) Database for northeast Tennessee.
- O'Lenic, E.A., Unger, D.A., Halpert, M.S., Pelman, K.S., 2008. Developments in Operational Long-Range Climate Prediction at CPC. Weather and forecasting 23, 496–515.

- Parsinejad, M., Yazdi, A.B., Araghinejad, S., Nejadhashemi, A.P., Tabrizi, M.S., 2013. Optimal water allocation in irrigation networks based on real time climatic data. Agricultural Water Management 117, 1–8. https://doi.org/10.1016/j.agwat.2012.10.025
- Pradhan, P., Fischer, G., van Velthuizen, H., Reusser, D.E., Kropp, J.P., 2015. Closing yield gaps: How sustainable can we be? PloS one 10, e0129487.
- Pretty, J., Bharucha, Z.P., 2014. Sustainable intensification in agricultural systems. Annals of Botany 114, 1571–1596. https://doi.org/10.1093/aob/mcu205
- PRISM, 2022. PRISM Climate Data [WWW Document]. PRISM Climate Group, Oregon State University.
- Purvis, B., Mao, Y., Robinson, D., 2019. Three pillars of sustainability: in search of conceptual origins. Sustainability science 14, 681–695.
- Rasmussen, R.K., 2010. Agriculture in History., Magill's Choice. Salem Press.
- Ricciardi, V., Mehrabi, Z., Wittman, H., James, D., Ramankutty, N., 2021. Higher yields and more biodiversity on smaller farms. Nature Sustainability 1–7.
- Richardson, K.J., Lewis, K.H., Krishnamurthy, P.K., Kent, C., Wiltshire, A.J., Hanlon, H.M., 2018. Food security outcomes under a changing climate: impacts of mitigation and adaptation on vulnerability to food insecurity. Climatic change 147, 327–341.
- Richetti, J., Boote, K.J., Hoogenboom, G., Judge, J., Johann, J.A., Uribe-Opazo, M.A., 2019. Remotely sensed vegetation index and LAI for parameter determination of the CSM-CROPGRO-Soybean model when in situ data are not available. ITC journal 79, 110–115.
- Ritchie, S., Hanway, J., Benson, G., 1986. How a corn plant develops. Spec. Rep. No. 48.
- Robertson, A.W., Kumar, A., Peña, M., Vitart, F., 2015. Improving and promoting subseasonal to seasonal prediction. Bulletin of the American Meteorological Society 96, ES49–ES53.
- Robertson, G.P., Vitousek, P.M., 2009. Nitrogen in agriculture: balancing the cost of an essential resource. Annual review of environment and resources 34, 97–125.
- Sabarina, K., Priya, N., 2015. Lowering data dimensionality in big data for the benefit of precision agriculture. Procedia Computer Science 48, 548–554.
- Saravi, B., Nejadhashemi, A.P., Jha, P., Tang, B., 2021. Reducing deep learning network structure through variable reduction methods in crop modeling. Artificial Intelligence in Agriculture 5, 196–207. https://doi.org/10.1016/j.aiia.2021.09.001
- Sarker, R., Ray, T., 2009. An improved evolutionary algorithm for solving multi-objective crop planning models. Computers and Electronics in Agriculture 68, 191–199. https://doi.org/10.1016/j.compag.2009.06.002

- Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the First International Conference of Genetic Algorithms and Their Application. pp. 93–100.
- Sharpley, A.N., Beegle, D., 2001. Managing phosphorus for agriculture and the environment. College of Agricultural Sciences, The Pennsylvania State University, University Park, PA.
- Singh, 2021. Evaluation and Performance of Different Irrigation Scheduling Methods and Their Impact on Corn Production and Nitrate Leaching in Central Minnesota (PhD Thesis). University of Minnesota.
- Singh, B., Chakraborty, D., Kalra, N., Singh, J., 2019. A tool for climate smart crop insurance: Combining farmers' pictures with dynamic crop modelling for accurate yield estimation prior to harvest. Intl Food Policy Res Inst.
- Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11, 341–359.
- Suresh, S., Sujit, P.B., Rao, A.K., 2007. Particle swarm optimization approach for multi-objective composite box-beam design. Composite Structures 81, 598–605. https://doi.org/10.1016/j.compstruct.2006.10.008
- Tamaki, H., Kita, H., S, K., 1996. Multi-objective optimization by genetic algorithms: a review. Diversity. https://doi.org/10.1080/02680939.2013.782512
- Tan, Z., Wang, H., Liu, S., 2021. Multi-stage dimension reduction for expensive sparse multiobjective optimization problems. Neurocomputing 440, 159–174. https://doi.org/10.1016/j.neucom.2021.01.115
- Thiaw, I., 2019. Loss of World's Arable Land Threat to 'Everything We Eat, Drink, Breathe', Speaker Says, as Second Committee Takes Up Sustainable Development [WWW Document]. UN. URL https://press.un.org/en/2019/gaef3519.doc.htm
- Tian, H., Lu, C., Pan, S., Yang, J., Miao, R., Ren, W., Yu, Q., Fu, B., Jin, F.-F., Lu, Y., others, 2018. Optimizing resource use efficiencies in the food–energy–water nexus for sustainable agriculture: from conceptual model to decision support system. Current Opinion in Environmental Sustainability 33, 104–113.
- Tian, Y., Cheng, R., Zhang, X., Jin, Y., 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine 12, 73–87.
- Tian, Y., Feng, Y., Zhang, X., Sun, C., 2022. A Fast Clustering Based Evolutionary Algorithm for Super-Large-Scale Sparse Multi-Objective Optimization. IEEE/CAA J. Autom. Sinica 1– 16. https://doi.org/10.1109/JAS.2022.105437
- Tian, Y., Liu, R., Zhang, X., Ma, H., Tan, K.C., Jin, Y., 2021a. A Multipopulation Evolutionary Algorithm for Solving Large-Scale Multimodal Multiobjective Optimization Problems. IEEE Trans. Evol. Computat. 25, 405–418. https://doi.org/10.1109/TEVC.2020.3044711
- Tian, Y., Lu, C., Zhang, X., Cheng, F., Jin, Y., 2020a. A Pattern Mining-Based Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems. IEEE Transactions on Cybernetics 1–14. https://doi.org/10.1109/TCYB.2020.3041325
- Tian, Y., Lu, C., Zhang, X., Tan, K.C., Jin, Y., 2020b. Solving Large-Scale Multiobjective Optimization Problems With Sparse Optimal Solutions via Unsupervised Neural Networks. IEEE Transactions on Cybernetics 1–14. https://doi.org/10.1109/TCYB.2020.2979930
- Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K.C., Jin, Y., 2021b. Evolutionary large-scale multi-objective optimization: A survey. ACM Computing Surveys (CSUR) 54, 1–34.
- Tian, Ye, Zhang, X., Wang, C., Jin, Y., 2020. An Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems. IEEE Trans. Evol. Computat. 24, 380–393. https://doi.org/10.1109/TEVC.2019.2918140
- Tian, Y., Zhang, X., Wang, C., Jin, Y., 2019a. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 24, 380–393.
- Tian, Y., Zheng, X., Zhang, X., Jin, Y., 2019b. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. IEEE Transactions on Cybernetics 50, 3696– 3708.
- Tyagi, A.C., 2016. Towards a second green revolution. Irrigation and Drainage 4, 388–389.
- UN, 2019. World population prospects 2019: Highlights (st/esa/ser. A/423).
- US EPA, 2020. Inventory of US greenhouse gas emissions and sinks. US Environmental Protection Agency, Office of Policy, Planning and Evaluation.
- van Dijk, M., Morley, T., Rau, M.L., Saghai, Y., 2021. A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050. Nature Food 2, 494–501.
- Vilayvong, S., Banterng, P., Patanothai, A., Pannangpetch, K., 2015. CSM-CERES-Rice model to determine management strategies for lowland rice production. Scientia agricola 72, 229– 236.
- Wang, X., Zhang, K., Wang, J., Jin, Y., 2021. An Enhanced Competitive Swarm Optimizer with Strongly Convex Sparse Operator for Large-Scale Multi-Objective Optimization. IEEE Trans. Evol. Computat. 1–1. https://doi.org/10.1109/TEVC.2021.3111209

- Waraich, E.A., Ahmad, R., Ashraf, M.Y., Saifullah, Ahmad, M., 2011. Improving agricultural water use efficiency by nutrient management in crop plants. Acta Agriculturae Scandinavica, Section B-Soil & Plant Science 61, 291–304.
- While, L., Hingston, P., Barone, L., Huband, S., 2006. A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation 10, 29–38. https://doi.org/10.1109/TEVC.2005.851275
- Whittaker, G., Färe, R., Grosskopf, S., Barnhart, B., Bostian, M., Mueller-Warrant, G., Griffith, S., 2017. Spatial targeting of agri-environmental policy using bilevel evolutionary optimization. Omega (United Kingdom) 66, 15–27. https://doi.org/10.1016/j.omega.2016.01.007
- Wortmann, C.S., Milner, M., Kaizzi, K.C., Nouri, M., Cyamweshi, A.R., Dicko, M.K., Kibunja, C.N., Macharia, M., Maria, R., Nalivata, P.C., others, 2017. Maize-nutrient response information applied across Sub-Saharan Africa. Nutrient Cycling in Agroecosystems 107, 175–186.
- Zhang, Q., Liu, W., Tsang, E., Virginas, B., 2009. Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Transactions on Evolutionary Computation 14, 456–474.
- Zhang, X., Tian, Y., Cheng, R., Jin, Y., 2018. A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization. IEEE transactions on evolutionary computation 22, 97–112.
- Zhang, Y., Gong, D., Sun, X., Guo, Y., 2017. A PSO-based multi-objective multi-label feature selection method in classification. Scientific reports 7, 1–12.
- Zhang, Y., Tian, Y., Zhang, X., 2021. Improved SparseEA for sparse large-scale multi-objective optimization problems. Complex Intell. Syst. https://doi.org/10.1007/s40747-021-00553-0
- Zheng, H., Ying, H., Yin, Y., Wang, Y., He, G., Bian, Q., Cui, Z., Yang, Q., 2019. Irrigation leads to greater maize yield at higher water productivity and lower environmental costs: a global meta-analysis. Agriculture, Ecosystems & Environment 273, 62–69.
- Zille, H., Ishibuchi, H., Mostaghim, S., Nojima, Y., 2018. A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation. IEEE transactions on evolutionary computation 22, 260–275.
- Zille, H., Mostaghim, S., 2017. Comparison study of large-scale optimisation techniques on the LSMOP benchmark functions, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, pp. 1–8.
- Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-report 103.